COMMODORE USERS

48355
K

Ahoy!

$2.50/ CAN. $3.00 AUG. 1985

A COMMODORE 128 PHOTO ODYSSEY!

WE GET INTO THE **C-128**
- INTERNAL HARDWARE
- BASIC 7.0
- DISPLAY TIPS

NOW YOUR COMMODORE *WILL* DO **WINDOWS**
MACINTOSH MAGIC FOR YOUR VIC AND 64 PROGRAMS

ART GALLERY
COMPUTER GRAPHICS AT THEIR PEAK

SCREEN COLOR AND **SCREEN DUMP** ROUTINES
• • • • • • • • • • •
ASSEMBLY LANGUAGE ADDRESSING

GIVE YOUR ORIGINAL GAME PROGRAMS MORE **BANG** WITH **SOUND EFFECTS** FOR THE VIC AND 64

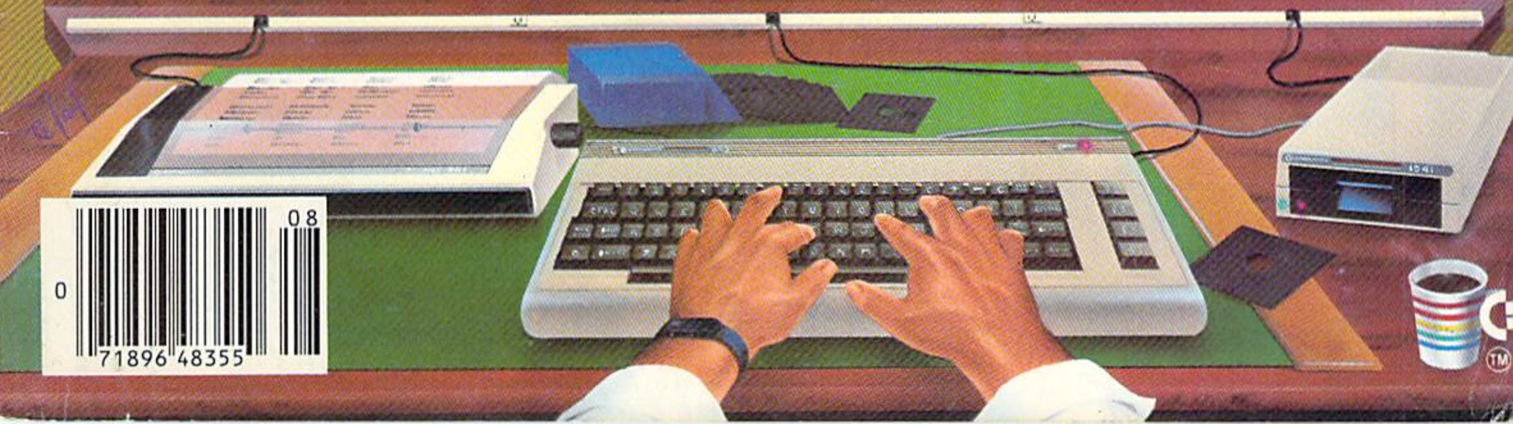
FIDGITS
A-B-C's MADE E-Z
• • • • • • • • • • •
GATORS N SNAKES
A SWAMP GAS!

FALLING NEEDLES AND SHARED BIRTHDAYS
REAL WORLD SIMULATIONS FOR THE VIC & 64

FORMATTER
PRINT STATEMENT WRITER
• • • • • • • • • • •
DISINTEGRATOR
EXPLOSIVE ML ACTION

REVIEWS OF
- FORTH-64
- MASTERDISK
- FELONY!
- MICRO MAIL

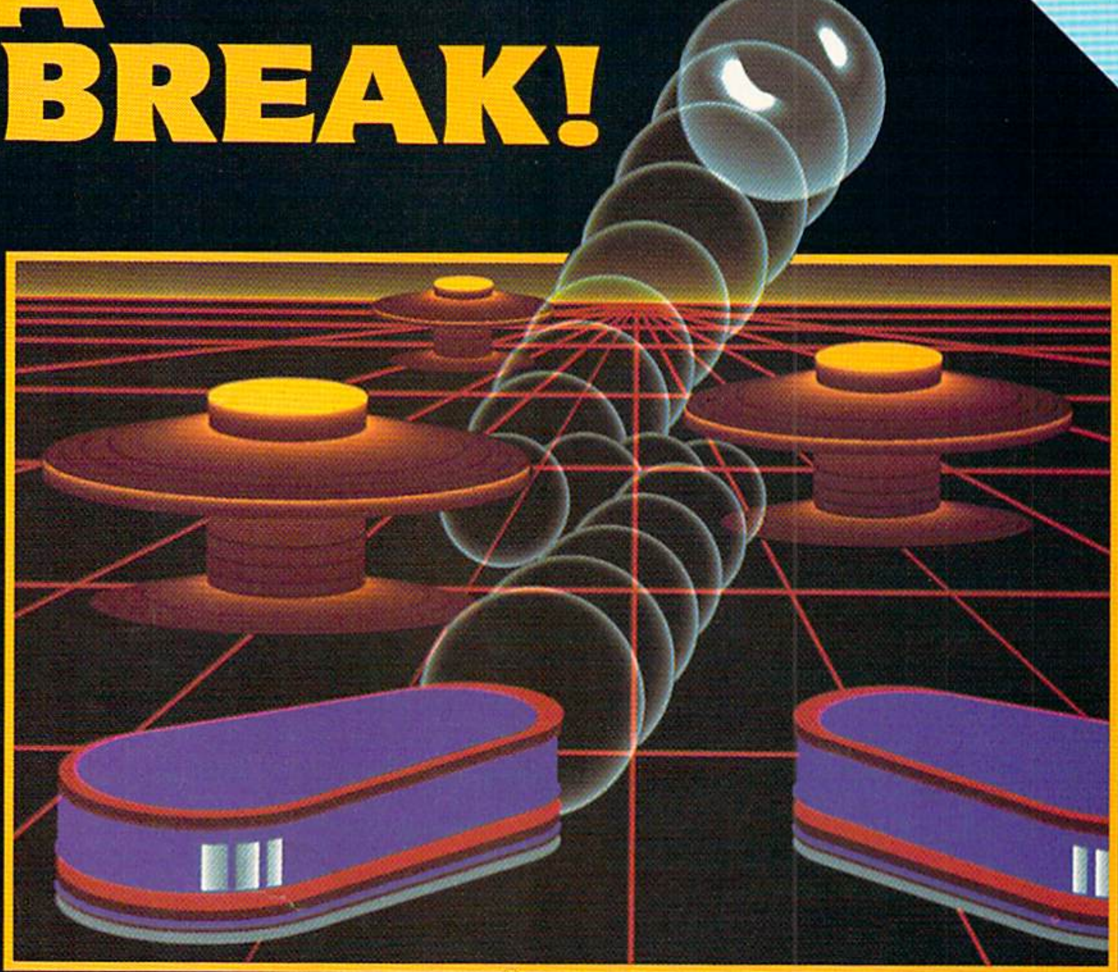
AND MORE FOR THE VIC AND 64...END



0 71896 48355 08

For Commodore 64™ Computers

TAKE A BREAK!



WITH NIGHT MISSION

PINBALL

You deserve the best. You've earned it. Now reward yourself with a session of Night Mission PINBALL, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put Night Mission PINBALL in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create *your own* custom modes. ■ So take a break with Night Mission PINBALL from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.



See your dealer . . .
or write or call for more information.

Order Line: 800 / 637-4983

subLOGIC
Corporation
713 Edgebrook Drive
Champaign IL 61820
(217) 359-8482 Telex: 206995

Reader Service No. 229

Ahoy!

CONTENTS

DEPARTMENTS

FEATURES

PROGRAMS

Cover art by James Regan; art and photos inside screens by Morton Kevelson, Armand Suarez, James Regan, and Javier Romero

Publisher
Michael Schneider

Editor
David Allikas

Managing Editors
Michael Davila
Robert J. Sodaro

Senior Editor
Tim Moriarty

Technical Editors
David Barron
Bob Lloret

Consulting Editors
Ben Bova
Morton Kevelson
Dale Rupert

Art Director
JoAnn Case

Production Director
Raoul Tenazas

Art Production
Christopher Carter
Mark Kammerer
Eve Griffin

Bulletin Board SYSOP
B.W. Behling

Circulation Director
W. Charles Squires

Advertising Director
Lynne Dominick

Director of Promotion
Joyce K. Fuchs

Controller
Dan Tunick

Managing Director
Richard Stevens

Advertising Representative
JE Publishers' Representative
6855 Santa Monica Blvd., Suite 200
Los Angeles, CA 90038
(213)467-2266
Boston (617)437-7628
Dallas (214)660-2253
New York (212)724-7767
Chicago (312)445-2489
Denver (303)595-4331
San Francisco (415)864-3252

ISSUE NO. 20 AUGUST 1985

Ahoy! (#8750-4383) is published monthly by Ion International Inc., 45 W. 34th St., Suite 407, New York, NY, 10001. Subscription rate: 12 issues for $19.95, 24 issues for $37.95 (Canada $26.95 and $49.95 respectively). Second class postage paid at New York, NY 10001 and additional mailing offices.

Postmaster, send address changes to *Ahoy!*, 45 W. 34th Street, Suite 407, New York, NY 10001. Direct all address changes or matters concerning your subscription to *Ahoy!*, P.O. Box #341, Mt. Morris, IL 61054. All editorial inquiries and software and hardware to be reviewed should be sent to *Ahoy!*, 45 W. 34th St., Suite 407, New York, NY 10001.

VIEW FROM THE BRIDGE

This month it begins—the most exhaustive coverage of the Commodore 128 you'll find in any computer periodical. Morton Kevelson starts out with an overview (and inner view) of the machine, its BASIC 7.0, and a guide to displaying 80 columns with a 1702 monitor. Next month it's on to the 1571 disk drive, along with a detailed look at the C-128's keyboard. But don't feel obliged to rush through the rest of this page—the article will wait. After all, you're the guest of honor at our *Opening Ceremonies for the Commodore 128.* (Turn to page 29.)

Another important beginning this month: our *Cadet's Column,* wherein Cheryl Peterson will start at square one for all beginning users of the Commodore 64. Future columns should provide interesting reading for advanced users as well (such as planned features on copy programs and computer furniture). (Turn to page 79.)

And while we're in a birthgiving mood, we'd like to announce the forthcoming debut of still another regular feature. See the information regarding our programming tips column on page 41.

That's enough new beginnings for now. Let's return to the tame stuff we present every month—the finest lineup of Commodore-related features and Commodore-compatible programs available between two covers.

Proving that through ingenious programming, Commodore computers can do almost anything that the higher-end home machines can, Thomas Bunker's *Windows on the VIC and 64* allows you to rope off portions of program screens for help messages and the like. (Turn to page 36.)

Marie E. Coon relieves you forever of the tedium of writing print statements with *Formatter,* a C-64 program that will generate program lines that exactly reproduce any screen full of text, graphics, or combination thereof. (Turn to page 13.)

Selectachrome by Simon Edgeworth (author of last month's *Printat*) provides a means of trying out different combinations of border, screen, and text colors without entering time-consuming commands. (Turn to page 17.)

Alfred J. Bruey's *Screen Dump Routine,* while not innovative, *is* short and handy. (Turn to page 46.)

In the games department, Barbara Schulak's *Disintegrator* compels you to defend eight underground silos from an enemy that descends with machine language speed. (Turn to page 17.) And James Sanders' *Gators N Snakes* offers graphics so realistic they're repugnant. (Turn to page 41.)

In the May edition of *Creating Your Own Games on the VIC and 64,* Orson Scott Card related an anecdote about some programs presented to him by Walter Meyers, a professor at North Carolina State. One of the very programs described in that column appears in this issue: *Fidgits,* an alphabet tutorial for the C-64. (Turn to page 39.)

Of course, Orson Scott's popular column appears in this issue as well. What the rather ridiculous title *Sputter Pop Hiss Chop Flap Crash Slurp Mash* boils down to is a tutorial on incorporating sound effects into your original game programs, including seven short sound effects programs for the 64 and five for the VIC 20. (Turn to page 18.)

Backing Orson Scott up this month is Bruce Bartlett, whose *Sound-A-Rama* offers a menu full of interesting sound effects to play on the 64. (Turn to page 27.)

This issue marks Dale Rupert's temporary departure from mainstream programming into a subject area seldom delved into by computer magazines: simulations of real-life phenomena on a Commodore 64, beginning with *Falling Needles and Shared Birthdays.* (Turn to page 53.) And your assembly language education continues with the first of two editions of *Commodore Roots* devoted to assembly language addressing. (Turn to page 73.)

This issue also marks the departure of managing editor Bob Sodaro, who has decided to trade staggering out of these offices at 9 p.m. every night for a lucrative freelance writing career. Who can figure that? Still, our very best to Bob.

And between these covers—our very best to you!

—David Allikas

GUEST EDITORIAL

By Cheryl Peterson

SoftCon, Atlanta, 1985. A dream of industry gurus singing the praises of an ever-expanding market and aisles of the latest software releases. A Commodore user's fantasy! Surely the "shakeout" couldn't touch a computer with a user base of several million.

The nightmare I found instead sent me home a day early, my dreams dashed by disheartening piracy figures, dire predictions, and the "why bother?" attitude of many exhibitors.

My bad experience is not important, but you should know that as a viable market, we've been written off. The opening panel discussion dismissed us as being nonexistent. Millions of home computerists were summed up and casually discarded like an old dog-eared hat.

"Oh, home users. There's no such thing. There are people who use computers at home for small business and educational purposes, but outside of that, forget it. There really isn't a "home" market; there never was. Let's face it, there's nothing left to sell. Most home computer owners have tossed their machines in a closet and given up."

Of course, this is paraphrased, but it accurately reflects the opinions of Mitch Kapor, president of Lotus Development Corp., and John Sculley, president of Apple Computers. Sculley pointed out that because so many Apples are used in schools, they aren't being relegated to closets.

Although Esther Dyson pointed out that Jack Tramiel would probably have felt differently, she also agreed with the others. As editor of a business-oriented newsletter, it wasn't surprising that she would so easily write us off. Fred Gibbons, a last minute substitute for Tramiel and president of Software Publishing Corporation, marketer of the *PFS* series, wouldn't stand up for us either. Of the four, though, he seemed most inclined to give us the benefit of the doubt.

And where was our "white knight?" His apology, posted in the press room, said only that he had another more pressing commitment. Tramiel, once the driving force of Commodore and now President of Atari, doesn't seem convinced that home computers are outdated. Instead, his newest ventures at Atari appear to be higher level computers at home market prices. (See *Scuttlebutt* section of April *Ahoy!* for details.)

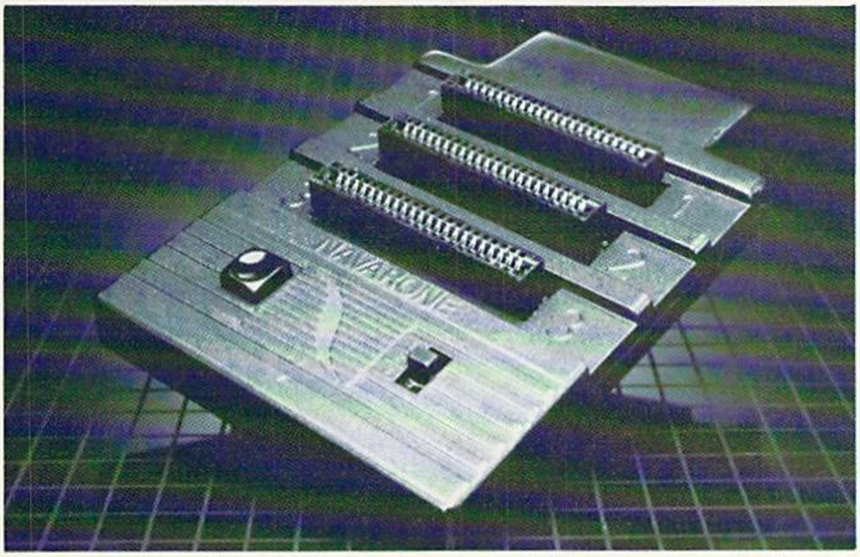
The dismal opening discussion wouldn't have caused me concern, except for the dearth of Commodore software among exhibitors. As a sales vehicle for software companies, SoftCon should have overflowed with vendors and programmers hawking their wares. It did! Unfortunately, everyone was driving an Apple or IBM.

Out of several hundred booths, less than a dozen had Commodore software. Of these, all but three were sell-

Continued on page 114

SCUTTLEBUTT

SPACE SIMULATION • MEDICAL DATABASE • TELECOM, SPREADSHEET TUTORIALS • DIGITIZER/CAMERA PACKAGE • CHEATSHEETS • EXPERT SYSTEM • OKIDATA, JUKI PRINTERS • $5 ASSEMBLER • C COMPILER • PROMAL UPGRADE • DVORAK CONVERTOR • BROWN BAG DATABASE/WP



Navarone's three-slot expander can save wear and tear on your cartridge port.
READER SERVICE NO. 200

THREE-SLOT EXPANDER

Navarone Industries, longtime manufacturer of a cartridge expander for the TI-99/4A (remember that one?), has made the same three-slot unit available for the C-64, enabling the user to shuttle between cartridges at the press of a button. Also included is a computer reset switch. Price is $24.95.

Navarone Industries, Inc., 19968 El Ray Lane, Sonora, CA 95370 (phone: 209-533-8349).

VIDEO DIGITIZER

Joining Computereyes (see April's *Scuttlebutt*), the first video acquisition system for the C-64, is Cardco's Digi-Cam. The package consists of digitizer (which connects directly to the C-64), software, cables, and a Panasonic monochrome camera with 12mm lens, all for $250. The image produced can be viewed on your monitor, stored on disk, transmitted by modem, or printed.

Acquisition time for a 320 x 200 dot screen image in five gray scales is 3 seconds. Included is a version of Cardco's *Paint Now* software to allow manipulation of the image. Word processor or light pen figures may also be applied.

Cardco, Inc., 300 S. Topeka, Wichita, KS 67202 (phone: 316-267-3807).

SPREADSHEET PRINTER

Sideways, which enables IBM and Apple users to print an entire spreadsheet vertically (and hence of any width desired), will be published for the C-64 and C-128 by Timeworks for use with all major brands of Commodore-compatible spreadsheets. Additionally, the program will be incorporated into Timeworks' own *Swiftcalc*. Price of the 64-compatible *Sideways* will be $29.95; *Swiftcalc* with *Sideways,* $49.95.

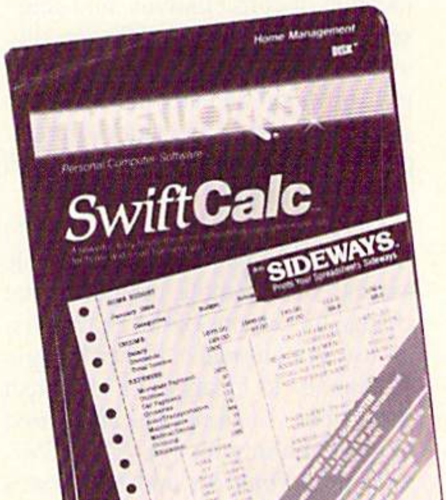
Timeworks, 444 Lake Cook Road, Deerfield, IL 60015 (phone: 312-948-9200).

EDUCATIONAL PROGRAMS

The 200-plus word vocabulary, speech synthesis capability, and animated letters of *Talking Teacher* combine to instruct two to eight year olds in the alphabet, spatial relationships, and placement of letters on the keyboard. For the C-64; $34.95.

Imagic, 2400 Bayshore Frontage Road, Mountain View, CA 94043 (phone: 415-940-6030).

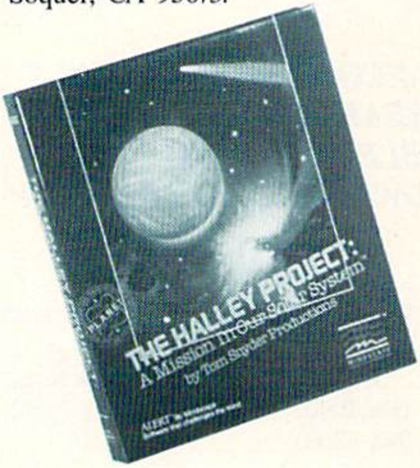
Speller's Duel enables two players to compete in a TV game show format, racing to spell words from five preprogrammed lists (for 5th, 6th,



Sideways *prints spreadsheet vertically.*
READER SERVICE NO. 201

7th, and 8th grades, plus Super Challenge). For the C-64; $24.95 plus $2.00 shipping.

J.M. Pizzuti Co., P.O. Box 1077, Soquel, CA 95073.



A real-time solar system simulation. READER SERVICE NO. 202

Three releases for the C-64 from Mindscape:

The Halley Project: A Mission in Our Solar System ($44.95) seeks players who qualify for the top secret project by completing a series of navigational tests in an authentic space simulation. The tests and obstacles help players master basic facts about the solar system, including details on Halley's Comet, gravity, atmospheric conditions, orbital motion, location of constellations, and eclipses. The program is a real-time simulation, meaning that every planet, star, or moon depicted moves at the same rate and in the same orbit or pattern as in our solar system. Atari version is included on flip side.

Mr. Pixel's Game Maker ($29.95) lets children play several example games, write simple programs that change the actions of the characters, or use conditional ("if-then") logic statements to change or establish game rules. The program can be used in conjunction with *Mr. Pixel's Programming Paint Set* and *Mr. Pixel's Cartoon Kit.*

Color Me ($29.95) allows young children to design their own pictures, coloring books, and stickers on the computer screen and then print them out. A *Color Me Supply Box* ($9.95), containing buttons, color printing papers, and adhesive-backed sticker paper, and *Color Me Picture Disks* ($9.95 each) will be sold separately.

Mindscape Inc., 3444 Dundee Road, Northbrook, IL 60062 (phone: 312-480-7667).

The *Success With Reading* book/software series will teach children reading via the Cloze technique, wherein students read and complete passages from which certain letters or words have been omitted. The four grade-specific packages (3 through 6), which sell for $49.95 each, must be used with the core package ($79.95). For the C-64.

Scholastic, Inc., 730 Broadway, New York, NY 10003 (phone: 212-505-3410).

In *Wally's Word Works,* students direct a bouncing wallaby along sentences, depositing words in the correct part-of-speech pockets. The $65.00 price includes teacher's diskette plus one of three student diskettes: *Wally's Workout* (elementary), *Pocket Pitfalls* (junior high), or *Rover's Revenge* (senior high to adult).

Sunburst Communications, Inc., 39 Washington Ave., Pleasantville, NY 10570 (phone: 800-431-1934 or 914-769-5030).



Color Me Picture Disks *(sold separately) feature Rainbow Brite, Shirt Tales, Hugga Bunch, and Tink!Tonk! characters. READER SERVICE NO. 203*

Turtle Tutor: A First Course in Logo, designed for use by both children and adults, teaches movement of the triangular "turtle" cursor by means of five games. Users move from elementary to more sophisticated commands, starting with matching and duplicating shapes displayed on the screen and progressing to using the full range of Logo commands to create original designs. Commodore Logo is required. For the 64; $44.00.

DLM Teaching Resources, One DLM Park, Allen, TX 75002 (phone: 214-248-6300).

Four C-64 releases for children 4 to 6 from Grolier Electronic Publishing, each $29.95:

The Story of Miss Mouse teaches the meaning of rhyming words through picture clues and color-coded typewriter keys.

Rhyme Land provides eight educational games that reinforce the concept of rhyme and provide 101 vocabulary words.

First Steps to Reading–Phonics Fun I and II teaches initial and final consonant sounds and their association with letter symbols, as well as the concept of spelling.

Exploring Your World consists of two programs: *Me and Others,* which explores the concepts of self and family, and teaches names of body parts and clothing; and *The Weather,* which covers vocabulary, concepts, and dress related to the weather.

Grolier Electronic Publishing, Inc., 95 Madison Ave., New York, NY 10016 (phone: 212-696-9750).

The price of *Lovejoy's Preparation for the SAT* was incorrectly listed in the April *Scuttlebutt* as $8.95. That, of course, is the price for the well-known paperbound volume. The price of the C-64 program of the same name is $69.95.

Simon & Schuster Electronic Publishing Group, 1230 Avenue of the Americas, New York, NY 10020 (phone: 212-245-6400).



Computerized aid—but not for $8.95.
READER SERVICE NO. 204

LOW-COST ASSEMBLER

Robin's Software has made their *IEA Editor Assembler,* consisting of disk plus looseleaf manual, available for $5.00 plus $2.00 postage (previous price was $19.95). As if that weren't generous enough, Robin's offers free help with 6502/6510 assembly language via their hot line: 612-829-0483 (5-10 p.m.).

Robin's Software, 10349 Zinran Circle, Bloomington, MN 55438.

C COMPILER

The first full C compiler for the 64, *Super C* is a complete development system that produces 6510 machine code. Its editor handles source files up to 41K in length. The linker accepts up to seven modules and the library supports standard as well as Commodore-oriented functions.

Abacus Software, 2201 Kalamazoo S.E., P.O. Box 7211, Grand Rapids, MI 49510 (phone: 616-241-5510).

EXPERT SYSTEM

Purportedly the first expert system for the C-64 and C-128, *XPER* allows the user to first build information into his database using the program's simple loading procedures, then use its searching techniques to guide himself through complex decision-making criteria. Full reporting and data maintenance capabilities are included.

Abacus Software, 2201 Kalamazoo S.E., P.O. Box 7211, Grand Rapids, MI 49510 (phone: 616-241-5510).

NEW GAME RELEASES

Activision will build QuickDrive, a software feature that speeds loading of a C-64 diskette by over 400%, into future game releases, including the following:

To obtain the fabled power and riches of the jeweled throne of *Alcazar: The Forgotten Fortress,* the player must traverse an enchanted land full of ancient castles, each containing tools indispensible to his quest, but guarded by griffins, genies, and the like.

In your attempt to halt the *Countdown to Shutdown* at the world's largest power plant, you direct your eight androids through the facility's 2000 rooms and deadly automated defense systems to find and repair the power core.

Participants in *The Great American Cross Country Road Race* (preannounced in the March *Scuttlebutt*) must plot a winning route, referring to a detailed US map and accounting for terrain, weather, and traffic. Gear shifting, fuel stops, engine maintenance, and quick driving reflexes are required along the way.

QuickDrive will also be incorporated into the previously announced *Web Dimension, Master of the Lamps, Rock N' Bolt,* and *The Music Studio.*

Activision, Inc., 2350 Bayshore Frontage Road, Mountain View, CA 94043 (phone: 415-960-0410).

Mr. Golf for the C-64/C-128 includes an 18-hole course, each hole with its own positioning of hazards like trees, water holes, and sand traps. Flight path of the ball is altered geometrically when hazards are encountered. A course design package uses icons to allow creating, deleting, and positioning greens, tees, hazards, and more. Joystick is required. Price: $19.95 tape, $22.95 disk.

APCAD Software, P.O. Box 2673, Ann Arbor, MI 48106.

Scheduled for November release, *The Kobiashi Alternative* is the first interactive fiction game based on Star Trek. The program features a personality database for all the most familiar characters. The player assumes the role of Kirk, leading the Enterprise on a quest to discover and neutralize the Bermuda Triangle of space. For the C-64; $39.95.

Simon & Schuster Electronic Publishing Group, 1230 Avenue of the Americas, New York, NY 10020 (phone: 212-245-6400).

The four latest war games for the C-64 from SSI:

Six Gun Shootout, a simulation of Old West-style man to man combat, lets one or two players choose up to 10 different scenarios of historical gunfighting, or fictional setups. Knives, tomahawks, rifles, shotguns, pistols, and 13 other types of weapons may be used. Price: $39.95.

Wings of War lets advanced wargamers choose from among 36 famous fighters or bombers, using up to 16 in each battle. Planes are rated for firepower, speed, maneuverability, and other factors. Five scenarios are included: air race, V-1 flying bomb intercept, night fighter, bomber formation intercept, and dogfight. Price of $39.95 includes two mapboard charts and three player aid charts.

Mech Brigade pits Soviet and

NATO forces against each other in a time period spanning from the 1980's to the 1990's. Over 45 different vehicles can be employed, as well as guided missiles, helicopters, and smoke. Four scenarios are included, plus the capability to create an infinite number of original ones. Price: $59.95.

Fighter Command recreates the air-war portion of the 1940 Battle of Britain, as the full 34-day campaign, three medium-length games, or a fast-paced Eagle Day assault. Each Brittish and German squadron is rated for pilot experience and morale, with recalculation after each mission. Player aid cards are included. Price is $59.95.

Strategic Simulations Inc., 883 Stierlin Road, Bldg. A-200, Mountain View, CA 94043-1983 (phone: 415-964-1353).

Datasoft will introduce five new Commodore 64 games this fall: *The Goonies* (based on the upcoming Steven Spielberg film), *Zorro, Alternate Reality, Pole Position II,* and *Elevator Action.*

Datasoft, 19808 Nordhoff Place, Chatsworth, CA 91311 (phone: 818-701-5161).

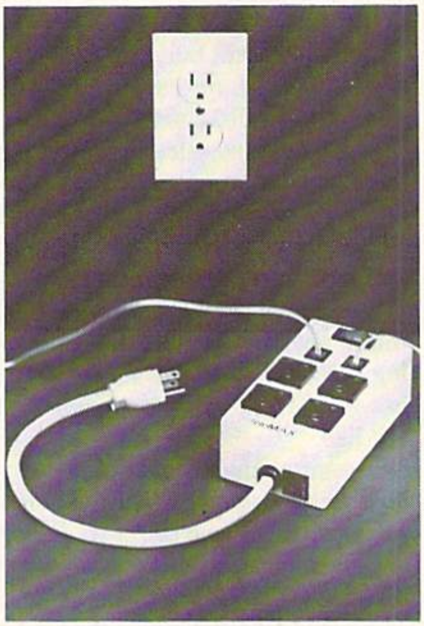
Mindscape will hold a contest in conjunction with *The Dolphin's Rune: A Poetic Odyssey,* in which the player assumes the role of a dolphin in order to locate and negotiate color currents and search seabeds for runes. Each rune unlocks hieroglyphics, encoded stanzas of dolphin lore. Nine such odysseys will yield nine stanzas of a poem. The first player to understand the poem, its dolphin lore, and the location of the secret wins the contest. Further contest details are available with the software, which is available (for the C-64) for $29.95.

Mindscape Inc., 3444 Dundee Road, Northbrook, IL 60062 (phone: 312-480-7667).

TELECOM NEWS

The Information Connection is aimed at teaching the beginner the fundamentals of tying into services like CompuServe and The Source. A tutorial simulates an online session wherein the user can practice searching for information; an actual telecommunications program permits the sending of electronic mail and messages to any online utility or BBS. Information may be dumped to any printer. Available in September for the C-64 and C-128; $59.95.

The Intelligent Choice in Data Communications, a brochure describing the USRobotics line of modems and telecommunications software, is available free by writing USRobotics, Inc., 1123 W. Washington Blvd., Chicago, IL 60607 or calling 1(800) DIAL USR.


TeleMAX: for spikes, surges, noise.
READER SERVICE NO. 205

The TeleMAX 4 telecommunications surge and spike suppressor and noise filter helps eliminate power, telephone, and telecommunication line disturbances that cause loss of data. Line protection is provided on the four AC receptacles and two RJ-45 RJ-11 compatible phone jacks. Price: $129.00.

Panamax, 150 Mitchell Blvd., San Rafael, CA 94903 (phone: in CA 415-472-5547; rest of US 1-800-472-5555).

LIGHT PEN, LIGHT PRICE

The Sure Light Pen for the C-64 and VIC 20 includes a high speed screen locator, a "game pak," and a tutorial for writing your own light pen programs. Price is $22.95.

Creative Enterprises, P.O. Box 2477, Big Bear City, CA 92314.

SMALL BIZ ACCOUNTING

Designed for the small business user, *Accountant, Inc.* comprises *General Ledger, Accounts Receivable, Accounts Payable,* and *Inventory* programs. It generates 35 different reports including an *Aging,* a *List of Payments,* and monthly statements of *Account Activities.* For the C-64; $59.95.

Softsync, Inc., 162 Madison Ave., New York, NY 10016 (phone: 212-685-2080).

FLOPPY PHYSICIAN

For the hypochondriac in all of us, *EMED,* the Electronic Medical Database for the C-64, contains information on over 460 diseases and 325 symptoms. It's possible to search/review and see all diseases for a symptom, or all symptoms for a disease. Included is a diagnostic program that lets the user select applicable symptoms, then displays the most probable diseases. On disk; $29.95.

DMC Software Co., Rt. 1, Box 364, Clinton, LA 70722.



Extra-wide Microline 193 supports spreadsheet applications of up to 233 characters per line, at a price of $200 higher than that of the narrower 192. READER SERVICE NO. 206

OKIDATA PRINTERS

The Microline 192 and 193 (wide-column) printers offer three different print modes, re-inking cartridge ribbon, an 8K buffer, hi-res correspondence quality mode, and a noise rating of 58 DbA. Priced at $499, the 192 comes with an adjustable pin feed mechanism; the 193 ($699) has a tractor feeder.

Okidata, 532 Fellowship Road, Mt. Laurel, NJ 08054 (phone: 609-235-2600).

LET YOUR FINGERS DO (LESS OF) THE WALKING

In an eight-hour day, claim the manufacturers of *DV Typer,* a DVORAK typist's fingers travel 1 mile, compared to 16 miles for a typist on the standard QWERTY keyboard. Hence their program that converts the C-64 keyboard to the DVORAK layout via software and durable overlays with dual (DVORAK and QWERTY) markings. *DV Typer* keeps records of WPM progress for one or more individuals learning DVORAK, has a QWERTY typing option, and can be exited so that other programs can be used with the DVORAK layout. C-64 disk, keycap overlays, and 22-page typing manual are $34.95.

Prelco, 14202 Galy Street, Tustin, CA 92680 (phone: 714-544-3041).

BROWN BAG SOFTWARE

The *Brown Bag* integrated database and word processor comes packaged in a lunchbag-style package to stress its no-nonsense, easy-to-use nature and its brown bag price of $49.95. Features include online help, merging of text and data between modules, and eight commonly used database templates.

Software Resource Group, Inc., 1095 Airport Road, Minden, NV 89423 (phone: 702-782-9731).

PERFECT DRAW

The *Perfect Draw* graphics design software will allow users of the *Perfect Writer* word processor to create and add a variety of illustrations and stylized text to their documents, with up to 128 colors for the C-64 and 128. The program is compatible with input devices such as joysticks, graphics tablets, and light pens. $50.00.

Thorn EMI Computer Software, Inc., 1881 Langley Ave., Irvine, CA 92714 (phone: 714-261-6600).

PROMAL UPGRADE

PROMAL Version 1.1, available to registered owners of Version 1.0 (reviewed in the June '85 *Ahoy!*) for $5.00 plus $2.50 shipping, corrects various reported problems and adds an ML subroutine interface and methodology for accessing Kernal routines, a diskette utility for backing up files, and support for the RS-232 interface, relative files, and assorted functions like ABS, SIN, COS, TAN, and ARCTAN. Price is $99.95 plus $5.00 shipping.

Systems Management Associates, Inc., 3325 Executive Drive, P.O. Box 20025, Raleigh, NC 27619 (phone:

919-787-7703).

CHEATSHEETS

Last time we reported on *Leroy's Cheatsheets* (Nov. '84 *Scuttlebutt*), there were 33. Seems Leroy had a few more up his sleeve: the current total of keyboard overlays available for assorted C-64 programs is 62. The most recent releases are *Fleet System 2, Sky Travel, Word Pro 64, Write Now!,* and *PractiCalc II.* Price is $3.95 each, plus $1.00 shipping.

Cheatsheet Products, P.O. Box 111368, Pittsburgh, PA 15238 (phone: 412-781-1551).



More **Cheatsheets** *from Leroy.*
READER SERVICE NO. 207

SPREADSHEET TUTORIAL

The latest in Grolier's series of entry-level, self-teaching productivity packages (see also *The Information Connection* on page 10), *Educalc* ($49.95) is designed to teach the beginning computer user to create and apply spreadsheets to everyday activities in homes and schools. Included is a self-instruction tutorial, pre-formatted practice template and menu-driven software tool that lets the student develop his own spreadsheets without memorizing commands. Available for the C-64 in September.

The *Educalc Template Disk* ($19.95), sold separately, includes 10 application templates preformatted with labels and formulas.

Grolier Electronic Publishing, Inc., 95 Madison Ave., New York, NY 10016 (phone: 212-696-9750).

MORTON THE K SPEAKS!

Ahoy! readers planning to attend the M.A.R.C.A. Commodore Computer Fair in Valley Forge, PA (July 26-28) won't want to miss Morton Kevelson speaking on printer interfacing (3:00 Friday and 1:00 Sunday). See page 101 for full details on the show.

FORMATTER

An Automatic Data Creator for the C-64

By Marie E. Coon

If the tedium of writing print statements to format that special game screen, menu, or other display has you climbing the walls, *Formatter* is the program for you! In just seconds, *Formatter* will generate new program lines that will exactly reproduce any screen full of text, graphics, or combination of the two that you devise, including all screen and character colors, even multicolor and extended background colors. In addition, *Formatter* makes either the full regular character set or the full upper and lower case set of 255 characters available for reprogramming, and will then create character data statements for all redesigned characters. You can create new lines for up to about 20 screens, and successively view them. A deletion feature allows you to delete the lines for any or all of the screens you have devised, or delete the *Formatter* program, leaving just your new lines for saving or appending. Finally, since one of my pet peeves is having to stop everything and search madly through a stack of magazines to find a forgotten program instruction, I have attempted to include all the instructions necessary to successfully operate *Formatter* within the program itself.

Because the memory requirements of *Formatter* plus the new program lines necessary to reproduce a reasonable number of screens is more than can be accommodated by the default VIC-II chip setting, some juggling of memory usage was necessary. To provide maximum space for BASIC programming while still having custom characters plus a screen for formatting and a screen for reprogramming, the latter operations were moved to the top of RAM memory, i.e., Bank 2, which starts at memory location 32768. An eight page block of memory from 32768 to 34815 is reserved for 256 programmable characters. A screen for formatting is located from 34816 to 35839 and a screen for reprogramming characters is located from 35840 to 36862. Memory locations 36863 to 40959 contain the character ROM image and cannot be used for screens or reprogrammable characters. An additional four page block of memory from 31744 to 32767 is reserved for storage of the color RAM for the formatting screen. The default screen which begins at location 1024 is also used for some of *Formatter*'s display features, leaving 29K free for BASIC programming.

Formatter is essentially a collection of subroutines that are called directly or indirectly by the main menu beginning at Line 400. To provide for a user-controlled delay with subsequent choices of action, the WAIT statement has been used throughout rather than the more familiar GET A$. This is because WAIT often requires one less program line, directly provides a numerical value, and does not create strings that can contribute to garbage collection problems.

Menu choice 1 lists the different color and switching options available on the formatting and reprogramming screens via the function keys. Because there are more choices than keys, some of the keys perform dual duty. This may be confusing at first, but once the differences between multicolor and extended background color modes and how they work is understood, there should be no problems in achieving the colors you want. Just remember that the border color will not change when you are in either extended background color or multicolor mode, and that these two modes are mutually exclusive—they cannot both be "on" at the same time. Also, they are turned off by pressing the same function key that turns them on. (Refer to the *Commodore 64 Programmer's Reference Guide* for a discussion of these features and how they operate.) The color numbers for the screen, border, and other color registers are cycled through their sixteen values by the use of the true/false test of a relational operator. A relational operator has a value of −1 when it is true, such as when SC in Line 200 is less than 16 in the expression SC<N where N equals 16. If SC is equal to or greater than 16, then the relational operator is false and has a value of zero. Therefore, the equation SC=−SC*(SC<N)+H where H equals one will cycle the value of SC from zero to 16. (See Lines 200 through 300 for more examples.)

The f4 key switches character sets back and forth between the ROM image and RAM. The somewhat uncommon BASIC numeric function ABS is used to toggle the character set value between 33 and 37 in Line 381. ABS strips the sign from a numeric value, leaving only the absolute value of that number. Therefore, O in the equation O=ABS(O-4) has a value of either zero or four, so that the expression 33+O will be either 33 or 37.

F6 switches directly back and forth between the formatting and reprogramming screens without going through the menu. The f8 choice flashes in red to remind you that it will call up the menu from either screen whenever you wish to make a different menu selection.

A second instruction screen may be called by pressing the space bar. It details how to change all characters printed on the format screen in a certain color to another color. Say you have some lettering in blue and you want to see how it would look in green. First change the cursor to blue by pressing the CTRL key and the 7 key together. Then press the CTRL key and the letter C key

together. The cursor will disappear. Now press the CTRL key and the 6 key together. Instantly, all of the blue lettering will change to green.

Menu choice 2 transfers program control to the formatting screen. It first appears as a pale grey screen with a white border and a flashing dark grey cursor in the home position. This is actually the real cursor that has been made visible by the machine language subroutine starting at location 49980. This subroutine also controls all of the keyboard input. When a function key is pressed, it stores the ASCII number of the key at zero page location 253 and then returns program control to BASIC. BASIC PEEKs the function key number and then selects the corresponding subroutine (program Lines 445 through 460). The character colors, reverse characters, and upper and lower case are controlled by the CTRL, COMMODORE, and SHIFT keys just as they are in direct mode. Also, all character keys have been set to automatic repeat. As its name indicates, the formatting screen is where the major screen design work takes place.

Program control is switched to the character reprogramming screen by menu choice 3. The most obvious feature of this screen is that it is divided in half across the middle. The upper half is colored an unchanging grey and contains some brief reprogramming instructions. This is where custom character designing takes place. The bottom half reflects the same screen and character colors as are on the formatting screen. This split screen is achieved by a machine language raster interrupt subroutine starting at location 49860. Also, as on the formatting screen, all keyboard input is controlled by the cursor subroutine at location 49980. Instead of switching character sets, however, the f4 key calls forth the reprogramming routine which starts at Line 150. To custom design a character, simply type the character you want to reprogram anywhere on the bottom half of the screen and then press f4. Immediately, an eight by eight grid of cyan zeroes and red asterisks will appear in the upper right quadrant of the screen, with the asterisks arranged in the pattern of the character you typed. The cursor will be flashing over the upper left corner of the grid and will respond to the screen editor directional keys and character keys, but not to the space bar, CLR, HOME, INST, or DEL. It is also confined to the grid. To redesign the character, just type in a different arrangement of zeroes and asterisks. The zeroes will always print in cyan and the asterisks in red so that your design will clearly stand out. To input your pattern, press any function key. The previously typed character will instantly change to your design. Further changes can be made simply by repeating this process until you are satisfied. A complete character set of 255 characters can be reprogrammed in this manner. Use the bottom half of the screen as a scratch pad to build multicharacter designs, or to experiment with character colors and arrangements for incorporation in full screen designs on the formatting screen.

Menu choice 4 jumps to the BASIC print statement subroutine beginning at Line 500. New print statements are generated by a combination of two machine language subroutines and BASIC. The first machine language subroutine establishes some markers and variables, and the second scans the formatting screen and POKEs the color, spacing, and character information for each screen line into a new program print statement. New lines are formulated at the top of the default screen and then added to the program via line 560. This line fills the keyboard buffer with RETURNS and then stop the program. The program pops in and out of direct mode to add each new line and to modify the controlling variables. New program lines start with Line 3002 and are incremented by two. The value for the next new line is held by variable D in Line 390. It is modified accordingly with each running of this subroutine.

The new program lines for each screen start with a preliminary line which calls a subroutine at Line 60002 that displays the starting line number for that screen. The second new line establishes the character set and the screen, border, and character colors. A third new line is generated if the new screen is in either multicolor or extended background color mode. Each screen's new lines end with a line which provides for a user-controlled delay.

The new screen display subroutine called by menu choice 5 starts at Line 1600. It first explains how to advance the display from screen to screen, or how to return control to the formatting screen or menu. Since any new screen is actually displayed on the formatting screen, any work in progress on this screen will be erased by running this subroutine. If you wish to view a previous screen(s) and also save your current work, first generate print statements for the unfinished screen and then run the display routine.

The deletion subroutine for menu option 6 begins at Line 720, and offers four deletion selections. Choice 1 will delete all of the new screen lines, choice 2 will delete all of the new screen lines from a particular line that you select, and choice 3 will delete all of the lines between two user-designated lines. To delete the lines for a particular screen, first get the screen's starting and ending line numbers from the display section. (The ending line number of a screen is two less than the starting line number of the next screen.) These options all use Line 560 in a variation of the print statement subroutine described above to delete the designated lines. Choice 4, to delete all but the new lines, employs a very different method which begins at Line 900. Actually, deletion of the old program is more apparent than real. The machine language subroutine starting at location 49152 scans through the BASIC program looking for the memory location of Line 3000. It then revises the BASIC pointers at locations 43, 44, 641, and 642 to indicate this location as the beginning of BASIC text. Now, only those lines from Line 3000 on can be LISTed, RUN, or SAVEd on tape or disk. The new lines will be automatically relocated to the default beginning of BASIC by the C-64 upon SAVing and reLOADing. The original *Formatter* program can

FREE!

BONUS GIFT WITH ANY SOFTWARE ORDER — AN EXCELLENT PUBLIC DOMAIN VERSION OF AN ARCADE HIT GAME ON ITS OWN DISK.

ARCADE

Archon	28.95
Archon II Adept	28.95
Beachead II	25.95
Bruce Lee	24.95
Castles of Dr. Creep	20.95
Championship Loderunner	24.95
Computer Quarterback	28.95
F15 Strike Eagle	23.95
Flight Simulator II	35.95
Impossible Mission	23.95
International Soccer	19.95
Kennedy Approach	23.95
Micro League Baseball	28.95
Statistics Disk (1984)	13.95
Mig Alley Ace	23.95
Night Mission Pinball	21.95
One On One	28.95
Pitstop II	27.95
Questron	27.95
Raid Over Moscow	27.95
Solo Flight	23.95
Strip Poker	20.95
Tournament Tennis	17.95
Zaxxon	24.95

ADVENTURE AND MIND GAMES

Amazon	23.95
Adventure Construction Set	37.95
Adventure Master	27.95
Below the Root	18.95
Bridge 4.0	17.95
Cutthroats	28.95
Dragonworld	23.95
Fahrenheit 451	23.95
Felony	23.95
Hitchhikers Guide	25.95
Infidel	25.95
Monty Plays Scrabble	27.95
Murder By The Dozen	23.95
Party Quiz	49.95
Party Quiz Refills	17.95
Rendevous With Rama	23.95
Robots of Dawn	27.95
Sargon III	34.95
Seven Cities of Gold	28.95
Starcross	26.95
Suspended	26.95
Swiss Family Robinson	18.95
Trivia Fever	27.95
Ultima III	39.95
Witness	25.95

EDUCATIONAL PROGRAMS

Agent USA	21.95
Cave of the Word Wizard	24.95
CBS Success w/Math Series	ea. 18.95
Evelyn Wood's Dynamic Reader	35.95
Harcourt Brace Computer SAT	54.95
Kidwriter 19.95 Memory Manor	19.95
Muppet Learning Keys	49.95
Spell It	33.95
Spellicopter	27.95
Story Machine	22.95
Word Attack	33.95

WORD PROC.-DATE BASES-SPREADSHEETS

Batteries Included Home Organizers	
Checkbook	20.95
Electronic Address Book	20.95
Home Inventory	20.95
Mail List	20.95
Photos/Slides/Movies	20.95
Recipes	20.95
Batteries Included Home Pak	34.95
Calc Result Advanced	69.95
Calc Result Easy	35.95
Complete Personal Accountant	54.95
Consultant (Delphi's Oracle)	68.95
Cut and Paste	36.95
Financial Cookbook	35.95
Fleet Systems	56.95
Magic Desk I	35.95
Mirage Data Base Manager	67.95
Mirage Word Processor	56.95
Net Worth	55.95
Paperclip w/Spellpack	82.95
PFS File	55.95
Practi Calc	34.95
Practi File	34.95
Superbase 64	52.95
Total Text 2.6	31.95
Total Mailing List & Labels	16.95

GRAPHICS

Animation Station	63.95
Doodle	28.95
Future House Light Pen	23.95
Moviemaker	37.95
Pen & Peripheral Vision	41.95
Print Shop	31.95

MISCELLANEOUS

Bulk Discs - Quantities of 10	12.95
Compuserve Starter Kit	21.95
Fast Load	27.95
Maxell Disks - Box of 10	16.95
Pro-Tech Vinyl Covers	5.35
Simon's Basic	38.95
Tac II Joystick	9.95
Vidtex	21.95
VIP Terminal	29.95

FLOPPY HOUSE

16 S. Chestnut Street, Box 308, Palmyra, PA 17078

HARDWARE AND PERIPHERALS

B.I. 80 Column Display	134.95
Cardco +G Interface	69.95
Commodore 801 Ribbons	7.95
Commodore Hardware	CALL
Indux Disk Drive	279.00
Okinate 10 w/Plug and Print	199.00
Okinate 10 Blk & Wht Ribbon	4.75
Okinate 10 Color Ribbon	5.25
Panasonic KXP1091 Printer	279.00
Prowriter 7500 Printer	229.00
Prowriter 8510 AP Printer	299.00
Prowriter 8510 Ribbons	5.95

LEROY'S CHEATSHEETS $2.95

Basic	Paperclip
Blanks (3)	Practicalc
Calc Result Adv.	Printers
Calc Result Easy	Epson RX80
Comal (1.4)	Gemini 10X
The Consultant	1525/801/
Doodle	Quick Brown Fox
Easy Script	Simon's Basic
Flight Simulator II	Speedscript
For the Beginner	Sprites Only
Hes Writer	Superbase 64
Logo Sheet 1	Terminals
Logo Sheet 2	Smart 64
The Manager	Vidtex
Multiplan	VIP
	Word Pro 3/Plus

SPECIALS

Cardco Numeric Keypad
32.95

Compuserve Starter Kit/Vidtex
39.95

S-Games (Formerly Hes Games)
11.95

TOLL FREE
1-800-633-8699

In Pa. Call
A.C. 717 838-8632

ORDERING & TERMS

- VISA & M.C. accepted with no charge in the continental U.S.A.
- $2.50 shipping charge on orders under $100.00
- Pa. Residents add 6% Sales Tax
- Prices subject to change w/o notice
- 48 hour shipping for cashier check - Money order - charge

Reader Service No. 226

be recalled by returning the pointers to their original values.

As mentioned previously, the new lines are formulated to display on the formatting screen located in Bank 2. To run your new lines on a screen located in another Bank, the POKE values in line 3000 must be changed accordingly. If you wish to use the default screen, Line 3000 may be deleted entirely and the POKE A (53272) values in subsequent lines either revised or deleted depending on whether the custom character set is used or not. In any case, the variables SC and BC should be deleted as they are only housekeeping values for *Formatter*'s use. Other lines, such as the GOSUB 60002, WAIT and IF G< >32 lines as well as Lines 60000 through 60004 should also be deleted unless you want them in your program.

Menu option 7 also uses a variation of the print statement subroutine to generate a data statement for each reprogrammed character. These statements begin with Line 60010 and increment by two. The machine language subroutine starting at location 49280 examines the custom character set and stores the screen code of each redesigned character in an array beginning at location 50432. BASIC PEEKs this number and prints it as the first number in each new data statement. Lines 60006 and 60008 establish a pair of nested FOR/NEXT loops to read and input the character data. As written, these lines will POKE the data into a character set located in Bank 2 beginning at location 32768. If you wish to use another location for your custom characters, then the value 32768 in Line 60008 must be revised to the beginning location of your set.

Menu choice 8 allows you to select which character set you wish to use for custom characters–regular or upper and lower case. Since the custom character set always occupies the same eight page block, both sets cannot be available for reprogramming at the same time. Also, the data statement subroutine will not make successive sets of data statements for both character sets. Statements written for the first set will be overwritten by the second set's statements. Therefore, if you decide to change sets but have already redesigned some characters of the other set that you wish to save, then make character data statements and save them on tape or disk before making the change.

The final menu choice, 9, exits the program to the default screen. I recommend using this exit, especially if you wish to LIST and/or revise your new lines. A RUN/STOP when the program is in Bank 2 will permit a readable listing, but a RUN/STOP RESTORE will result in an apparent limbo where little seems to work properly. Sometimes RUN plus RETURN will restart the program, but occasionally not. Best to avoid the whole problem.

There are only three more things you need to know about *Formatter*. First, you cannot use quotation marks in formatting a screen. The print statement subroutine has been programmed to ignore them and substitute a space in their place. This is because BASIC does not allow a quotation mark to be included in a print statement. If you wish to use what appears to be quotation marks, then program another character to serve that purpose. Second, if you use the INSerT key to position a line on the formatting screen, be sure to use the space bar or some other character to erase the quote mode characters that the INSerT key generates. These characters are not visible at first but will show up later when the new print statements are run. Lastly, *Formatter* will not pick up a reprogrammed space character if it is the beginning or ending character in a line. This is because of the way the print statement subroutine works. If the first character of a screen line is a space, then it is recorded as a cursor right, not a space. This continues until some other character is found and recorded. Spaces between characters are then recorded as spaces, but the spaces ending a line are simply forgotten.

A special note for Eastern House *Rabbit* users: the *Rabbit* uses at least one of the same zero page memory locations to establish its link with BASIC as is used by a couple of *Formatter*'s machine language subroutines. This will cause a serious program crash. Therefore, *Rabbit* link into BASIC must be removed by typing *K before RUNning *Formatter*. However, the *Rabbit* link may be reestablished to SAVE and LOAD either *Formatter* or new program lines created by it with no problems. ☐

SEE PROGRAM LISTING ON PAGE 101

SELECTACHROME

A Color Experimentation Utility for the C-64

By Simon Edgeworth

Do you write your own programs? If you do, it is important to use interesting and effective color combinations on the screen. Experimenting with different colors normally involves juggling with commands such as POKE53281,15 or PRINTCHR$(144). This can be both frustrating and time-consuming.

Selectachrome is designed to make color experimentation easy. It enables you to try out over 40,000 different combinations of border, screen, and text colors. When you have found a color scheme which you like, reproducing it in your own program is simple, because *Selectachrome* tells you exactly which BASIC commands to use.

Colors are changed by pressing the function keys. Fl changes the border, f3 changes the screen, and f5 and f7 change the two text colors. The shifted function keys step through the colors in reverse direction.

The text on the screen is continually updated to show the names of the current colors, and the corresponding BASIC commands.

Happy colorscheming! ☐

SEE PROGRAM LISTING ON PAGE 110

DISINTEGRATOR

An Arcade Action Game for the C-64

By Barbara Schulak

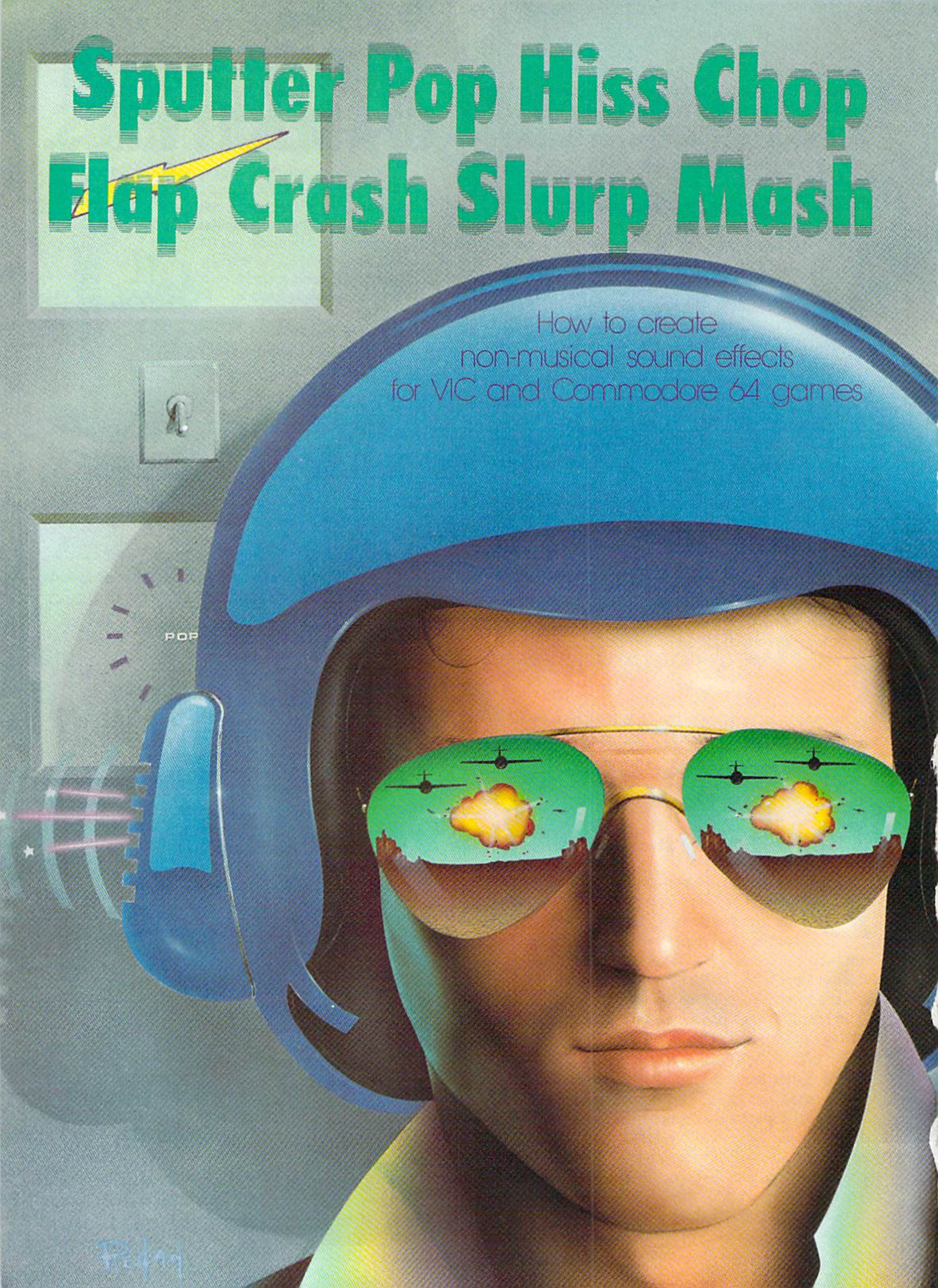
Disintegrator is an all machine langauge arcade game. The object is to protect your base from the invading enemy. For your defense, you have at your disposal eight underground silos from which you can launch bombs that will disintegrate the oncoming enemy. These bombs are activated by pressing the appropriate number key (1-8).

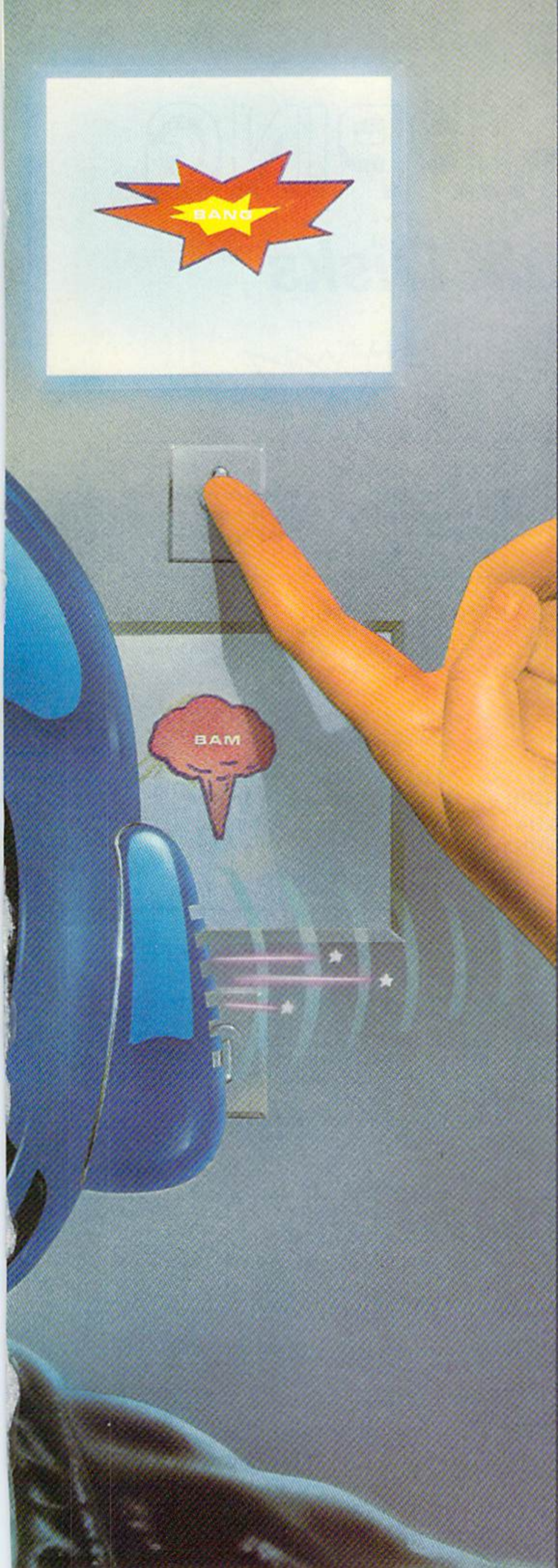
The enemy comes in waves of 15. Once all 15 ships have been successfully shot down, the level is complete and the next wave begins to attack at an increased rate of speed. There are nine levels in all. Pressing the space bar will pause the game. Pressing the space bar a second time will restart the game.

If the enemy successfully reaches your base, one of your "lives" is lost. After three hits, your base is destroyed, the score is displayed, and you may choose to play again by pressing "Y". Hitting the RETURN key starts the game. ☐ *SEE PROGRAM LISTING ON PAGE 98*

Sputter Pop Hiss Chop Flap Crash Slurp Mash

How to create non-musical sound effects for VIC and Commodore 64 games





James Regan

CREATING YOUR OWN GAMES ON THE VIC AND 64

There were a couple of guys in Orem, Utah, who used to sing along with the *Crazy Climber* game. Not humming, *singing*, in full operatic voices. It attracted a lot of attention, but the truth is that they were merely doing out loud what we all secretly do in our minds. The music gets to you.

You don't realize how much you need it until you play a machine whose sound is damaged or turned off. In a mall in Greensboro there's a *Crystal Castle* with one voice missing–drives me crazy. And the *Donkey Kong* in a mall in South Bend, Indiana, that has no music at all–well, I can't play it. I just can't seem to get the rhythm.

And that's just the music. Even more important are the sound effects. Instead of providing background or mood, they actually communicate to the player what is going on.

YOU CAN'T LOOK AT EVERYTHING

When you're playing you can't keep your eyes on everything. Even when you get that glassy, unfocused gaze that the best players get, there's still a limit to how many things you can pay attention to at the same time–with your eyes.

But your ears can pick up a lot of additional information. The shooting sound tells you that the computer got your "fire" message. The explosion tells you that you hit one of the bad guys. The much louder explosion tells you that the bad guys got something you were trying to protect.

There are beeps for picking something up and boops for dropping something, boinks for bumping into walls, hums to announce that the worst monster is about to come in and eat you, and sad little melodies to tell you that you're dead.

Sometimes the sounds are realistic–an engine sound for a car race game, for instance. The higher it rises in pitch, the faster you're going. The almost painful squeaking sounds in *Mario Brothers* tells you you're walking on ice.

Sometimes the sounds are nothing like the real world. But as you play the game, you learn which sound means which event. And from that moment on, you don't even have to *see* the event to know that it's taken place. The sound of the bouncing ball in *Mr. Do* tells you that it hasn't hit anything yet, so you know you can't throw it again.

The sounds you create for your game need to be distinct from each other, so that the player doesn't get confused about what has happened.

The sounds also should be interesting, sometimes

By Orson Scott Card

strange, sometimes comic, to make your game feel new. Don't you get tired of the same old shooting noises when you play other people's games?

MAKING SOUNDS ON THE 64

In the last two columns we've covered everything about sound production on the VIC and 64 except the noise generators. But it doesn't hurt to review. Here's a useful table of all the locations used for the various sounds.

Volume Control. A number from 0 (softest) to 15 (loudest) is POKEd into 54296. A value of 0 turns off all sound.

Voices. Three voices are available, each separately controlled. Voice 0 has seven registers starting at 54272; voice 1 has seven registers starting at 54279; voice 2 has seven registers starting at 54286. Starting with the voice's base address and adding the value shown, each of the registers are as follows:

Pitch Control: Base+0 and base+1. The first address contains the low byte, the second address the high byte of the pitch of the note to be played by that voice. For rough pitch control (where accurate melody doesn't matter), only the high byte need be used. For the voice to make any sound at all, however, one of the pitch (or frequency) registers must have a non-zero value.

Pulse Width: Base+2 and base+3. The first address contains the low byte and the second address the high nybble of the pulse width. These registers are only used if the pulse wave form is selected in the voice control register below; otherwise, these locations have no effect.

Voice Control Register: Base+4. Each bit of this register sends a different message to the sound chip. Each bit can be turned on by ORing it with a number or turned off by ANDing it with a number. As each function is described, the numbers to OR (turn on) or AND (turn off) are included. To use those numbers, you would use this formula:

```
POKE register, PEEK(register)AND turn off number
```

or

```
POKE register, PEEK(register)OR turn on number
```

Here are the functions of the voice control register:

(Gate): Begin executing the sound envelope (OR 1) or release the currently sustained sound envelope (AND 254). This has the effect of turning each voice on or off.

(Sync Bit): Synchronize with another voice; OR 2, AND 253. This is an advanced sound technique.

(Ring Modulation): If the triangle waveform is selected, add overtones from another voice; OR 4, AND 251. This is an advanced sound technique (see "SAWING WOOD" for an example of its use).

(Test Bit): OR 8, AND 247. Advanced sound technique.

(Triangle Waveform): Turn off all waveforms with AND 240, then OR 16.

(Sawtooth Waveform): Turn off all waveforms with AND 240, then OR 32.

(Pulse Waveform): Turn off all waveforms with AND 240, then OR 64. A pulse width must be selected at base+2 and base+3 when using this waveform.

(Noise Waveform): Turn off all waveforms with AND 240, then OR 128. The pitch has a less noticeable effect on the noise waveform, but the difference between a high-byte pitch of 255 and a high-byte pitch of 1 is very clear.

One of the waveforms must be selected before any sound can be heard. Since both the waveform and the gate are necessary to cause a sound, the two numbers can be combined and POKEd in at once. For instance, to gate on voice 0 with a sawtooth waveform, you could use

```
POKE 54276,33
```

Sound Envelope: Base+5 and base+6. The attack value, a number from 0 to 15, **times 16,** determines how quickly the volume rises from nothing to loudest when the note is first gated on; 0 is quickest, 15 slowest. The decay value, a number from 0 to 15, determines how quickly the volume falls from the peak volume down to the sustain volume; again, 0 is quickest.

The sustain value, a number from 0 to 15, **times 16,** is the volume at which the note is sustained until it is gated off; 0 is softest, 15 loudest. The release value, a number from 0 to 15, determines how quickly the volume falls from the sustain value to silence when the note is gated off; again, 0 is quickest.

The attack and decay are combined and POKEd into base+5. The sustain and release are combined and

POKEd into base+6. For instance, suppose we want an attack of 3, a decay of 9, a sustain of 5, and a release of 15. We would use this formula:

```
POKE base+5, (3*16) + 9:POKE base+6,
(5*16)+15
```

Sound envelopes apply regardless of the waveform selected. This allows some control of noises as well as musical tones.

MAKING SOUNDS ON THE VIC

The VIC makes its noises much more simply. Only five registers are used.

Pitch Control. The first three registers produce musical notes. Each register has its own waveform, which cannot be changed, and each register is exactly one octave higher than the register before. The lowest is at address 36784, the middle voice at address 36785, and the highest at address 36786.

The fourth register, at 36787, produces noise instead of a musical note, but it is used exactly the same way as the three musical note registers.

To select the pitch for each voice, including the noise register, you use a number from 0 to 127. 0 is the lowest and 127 the highest pitch. Remember, a number POKEd into 36784 is exactly two octaves lower than the same number POKEd into 36876.

To turn the voice on, you add 128 to the pitch before POKEing it in. So to play a C, which has a pitch of 97, you would use

```
POKE register, 97+128
```

If a register already contains the pitch you want it to play, and you merely want to turn it on, you can use

```
POKE register, PEEK(register) OR 128
```

To turn off the register without disturbing the pitch it is playing, you can use

```
POKE register, PEEK(register)AND 127
```

Volume Control. The volume is controlled at location 36878, and must be a number from 0 (silent) to 15 (loudest). The same location also controls the auxiliary color for the screen if you are using multicolor mode. If you aren't using multicolor mode, then no harm is done if you simply POKE the volume value into 36878. If you need to protect the auxiliary color code, then you will use:

```
POKE 36878,(PEEK(36878)AND 240)OR
volume
```

MANIPULATING THE SOUND WHILE IT'S PLAYING

With the music programs in the last two issues of *Ahoy!*, we have done nothing more than play simple music. We set the waveforms, envelopes, volume before the melody is played, and then changed the pitch from note to note, gating each note on, counting the duration, and gating it off when its time is up.

To create more elaborate sound effects, we don't play so many *different* notes, but each one lasts longer, and we manipulate pitch and volume (and, with the 64, waveform) *while the note is being played.*

Often, you will want to get rid of the 64's sound envelopes for sound effects (though just as often you will want to use envelopes). To get rid of the sound envelope, you simply use a value of 0 for the attack/decay register and a value of 15 for the sustain/release register. This has the effect of bringing the voice up to full volume as quickly as possible, leaving it at full volume as long as the note is held, and then shutting it off immediately as soon as the note is gated off. You will notice that this makes the 64 function much like the VIC.

To shape a sound effect, you first set up the note by selecting waveform, envelope, and so forth, along with the *initial* pitch and *initial* volume. Then you gate the sound on. As soon as the sound is playing, you begin manipulating it.

Changing Pitch. You can start the pitch high, then bring it down by POKEing successively lower numbers into the pitch control register, using a "backward" loop:

64 version:

```
FOR I=255 TO 5 STEP -10:POKE pitch high-
byte register, I:NEXT
```

VIC version:

```
FOR I=125 TO 5 STEP -5:POKE voice
register, I+128:NEXT
```

Notice that I used a step of 10 (5 in the VIC version). The higher the step value, the greater the change in pitch at each step in the loop. In this loop, the pitch would get lower in 26 increments.

Because the sound is not gated on or off between each pitch change, the listener usually can't hear the change from one pitch to the next. It sounds like a smooth, continuous swoop.

If the loop went from 5 to 125, in steps of 20, it would swoop upward very quickly; if it went from 5 to 125 in steps of 2, it would rise very slowly.

Or you can create a vibrato by alternating between two pitches, like this:

```
FOR I=0 TO 19:POKE pitch register,
60:POKE pitch register, 90:NEXT
```

By adding short delay loops—like FOR X=0 to 9: NEXT—between each pitch change, the vibrato could be made more distinct.

These pitch changes also can be used with noise waveforms on the 64 and the noise voice on the VIC, but the differences are much less clear. However, it does change the quality of the noise considerably.

MUST LIQUIDATE At Far Below Dealer Cost!

TOTAL Personal Computer System

FACTORY REFURBISHED!

GET ALL THIS AT ONE LOW PRICE!

64K Computer • 16 Hue Color, 5" Monitor • Disk Drive • Printer • Two Software Packages: "Easy Script" word processing and "The Manager" data base program!



Carries easily as a suitcase! Plugs into 115V outlet.

803 Printer

Sorry, we're not permitted to PRINT the famous brand-name. BUT, we CAN "tell all" if you call us TOLL FREE: 1-800-328-0609!

THE COMPUTER.

For business, education and home use! Thousands of programs are available.

Snap-on computer! 64K RAM, 20K ROM. Full-size typewriter keyboard. Upper and lower case letters, numerals, symbols, reverse characters. 2 cursor control keys, 4 function keys, programmable to 8. Music synthesizer with 3 independent voices, each with 9 octave range. Input/output ports accommodate . . . user, serial, ROM cartridge, joysticks, external monitor, phone modem.

Built-in disk drive! Intelligent high speed unit with 5¼" floppy disk recorder. 170K formatted data storage; 35 tracks. 16K ROM. Uses single sided, single density disk. Serial interface. Second serial port to chain second drive or printer.

Built-in color monitor! Displays 40 columns x 25 lines of text on 5" screen. High resolution. 320 x 200 pixels. 16 background, character colors.

Built-in ROM cartridge port! Insert ROM program cartridge. Multitude of subjects available in stores across the nation!

THE PRINTER.

Graphics quality 6 x 7 dot matrix printer. An ideal "fit" for this system . . . and included in the astonishingly low package price below!

Print method: Bi-directional impact dot matrix.

Character matrix: 6 x 7 dot matrix.

Characters: Upper and lower case letters, numerals and symbols. All PET graphic characters.

Graphics: 7 vertical dots — maximum 480 columns. Dot addressable.

Character codes: CBM ASCII code.

Print speed: 60 characters per second.

Maximum columns: 80 columns.

Character spacing: 10 characters per inch.

Line feed spacing: 6 lines per inch in character mode or 8 lines per inch selectable. 9 lines per inch in graphics mode.

Line feed speed: 5 lines per second in character mode. 7.5 lines per second in graphics mode.

Paper feed: friction feed. (Tractor feed optional.)

Paper width: 4.5" to 8.5" width. Up to 10" with optional tractor feed.

Multiple copies: Original plus maximum of two copies.

Dimensions: 13"W x 8"D x 3¼"H. **Wt.:** 6½ lbs. **Power:** 120V AC, 60 Hz.

THE SOFTWARE.

"Easy Script" One of the most powerful word processors at any price! Cut re-typing, create documents from standard paragraphs, do personalized letters, see and change a document before it is printed. Instruction manual has extensive training section that simplifies use . . . even for someone who has never used a computer or word processor before!

"The Manager" A sophisticated database manager for business or home use. Business uses: accounts payable/receivable, inventory, appointments, task manager. Home uses: mailing lists, home inventory, recipes, collection organizer, investment tracking, checkbook balancing. School uses: research article index, gradebook.

Compatible with Commodore Computer Systems (Not included in package price.)

JOYSTICKS (Set of 2)
Give you the fun of playing a wide selection of games.
Mfr. List: $59.90 pr.
Liquidation Price .... **$18 pr.**
Item H-577-63622-01, S/H: $6.00 pr.

64 MODEM
Access new worlds of information with your computer.
Mfr. List: $124.95
Liquidation Price **$44**
Item H-577-63646-00, S/H: $4.00

TOTAL

Personal Computer System Now Available At FAR BELOW dealer cost!

TOTAL PACKAGE PRICE **$488**

Item H-577-64011-02
Shipping, handling: $24.00

Other popular brands of computer peripherals available at liquidation prices. For information, Call TOLL FREE: 1-800-328-0609.

VISA MasterCard AMERICAN EXPRESS DINERS CLUB

Credit card members can order by phone, 24 hours a day, 7 days a week.

Toll-Free: 1-800-328-0609

TeleCheck Your check is welcome! No delays when you pay by check!

C.O.M.B. CO.®
Authorized Liquidator
14615 28TH AVENUE NORTH
MINNEAPOLIS, MINNESOTA 55441-3357

SEND ME THE ITEMS I HAVE LISTED BELOW
Price subject to change after 60 days. Sales outside continental U.S. are subject to special conditions. Please call or write to inquire.

Item No.	How Many	Item	Ship/ Handl.	Price

C.O.M.B. CO.® Item H-577
14615 28th Ave. N./Minneapolis, MN 55441-3357

Send the items indicated at left. (Minnesota residents add 6% sales tax. Please allow 3-4 weeks for delivery. Sorry, no C.O.D.)

☐ My check or money orders is enclosed. (No delays in processing orders paid by check, thanks to TeleCheck.)

Charge: ☐ MasterCard® ☐ VISA ☐ Am. Ex. ☐ Diners Club

Acct. No. ______ Exp. ______

PLEASE PRINT CLEARLY

Name ______

Address ______

City ______

State ______ ZIP ______

Phone ______

Sign here ______

Sensational Prices!
. . . On Our Most Popular Items!

Introducing . . .

THE 99¢ DISKETTE

Are you paying too much for diskettes? Try our first quality, prime, 5¼" diskettes (no rejects, no seconds) at these fantasic sale prices and save, save, SAVE! Disks are packaged in boxes of 50; each box contains 5 shrink-wrapped 10-packs that include diskettes in sleeves, labels, and write-protect tabs.

Each diskette is certified to be 100% error free and comes with a lifetime warranty (if you have a problem, we'll replace the diskette). All diskettes include hub reinforcement rings and write-protect notch.

All diskettes are double density and work in either single or double density drives.

SS, DD Diskettes, Box of 50
32391 — $49.50–99¢ ea.!

DS, DD Diskettes, Box of 50
32403 — $64.50–$1.29 ea.!

LOWEST PRICES!

$

We can offer you some of the lowest prices in the country on the most popular printers, monitors and interfaces. Our normal prices are already low, but to make sure you get the best deal you can, we will also meet most competitive prices in this publication when placed on an equal basis (Remember—we don't charge for use of your credit card, impose excessive shipping fees, or use any other hidden extras to boost the price you pay. Due to the rapid change in prices in the computer industry, we can only meet prices at the time you place your order; we cannot adjust prices on items ordered or shipped on an earlier date.) Another plus for charge card customers—your charge card is billed at time of shipment only for the items shipped—no early billing, no long wait for the merchandise you already paid for.

star
EPSON
BMC
AXIOM
CARDCO

★ THE BEST PRICES ★ THE BEST SERVICE ★
WHY SHOP ANYWHERE ELSE?

COMPUTERFACTS™ from SAMS

Technical Service Data for Your Computer and Peripherals



Sams introduces COMPUTERFACTS™ . . . packets of information that reveal the inner workings of major brand micros, monitors, printers and disk drives. Includes schematic wiring diagrams, parts lists, disassembly instructions, troubleshooting techniques, and other repair data.

33477 Commodore 64 Computer
33513 VIC-20 Computer
33551 Commodore Plus/4 Computer
33481 Commodore 1541 Disk Drive
33496 Commodore 1525 Printer
33508 Commodore C16 Computer
33546 Commodore 1701 Monitor

All packets $17.95 each

The Most Complete Disk Drive You Can Buy

Includes Word Processor, Spread Sheet, and Data Base Manager

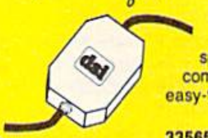
INDUS GT ONLY $259

State-of-the-art disk drive for C-64 and VIC-20 features LED sector indicator, electronic write-protect, built-in wedge, and operating speeds up to 400% faster. Package includes cables and power supply, carrying case that doubles as a disk storage file AND software starter package featuring word processor, spreadsheet, and database manager program!
FULL YEAR PARTS AND LABOR WARRANTY!!

33443 Indus GT Disk Drive — ONLY $259

GRAPHICS PRINTER INTERFACE

PPI with graphics



The PPI (Parallel Printer Interface) from DSI is a high-performance graphics printer interface for your Commodore 64 or VIC-20, offered at an unbelievable price! The PPI fully emulates a Commodore printer so that all software written for Commodore computers and printers can be run with the PPI and most any parallel printer. Print true Commodore graphics and alpha-numerics as well as any special character sets offered by your specific printer. Works with Star, Epson, BMC, and many more. Comes complete with all necessary cables for quick installation and includes easy-to-follow user's manual.

33565 Parallel Printer Interface — $44.95

CARTRIDGE EXPANDER

NAVARONE

Plus $22.95 FAST LOAD OFFER!

Stop wearing out your computer and cartridges by endless swapping! New design cartridge expander features two cartridge slots. Select either cartridge, or the "off" position if no cartridge is to be used. Reset the computer independently of the power switch. Cartridge slots are vertical for easy access—no blind fumbling behind the computer.

34220 2-Slot Cartridge Expander — $19.95

The cartridge expander is a great companion for the Epyx Fast Load cartridge—you can keep it permanently installed plus have a slot free for other cartridges!

34216 Fast Load Cartridge (Sug. Retail $34.95) — $29.95

Fast Load ONLY $22.95 with purchase of cartridge expander!

DUST COVER and "EVERYTHING BOOK" INTRODUCTORY SPECIAL

Get to know us by ordering this great dust cover for your VIC-20 or Commodore 64 and our catalog, "The Everything Book for the Commodore C-64 and VIC-20 Home Computers," for $2.95 (no extra shipping or handling charges). Cover is antistatic, translucent, 8 gauge vinyl sewn to our exacting standards with reinforced seams. Discover the savings and easy shopping available from Tenex Computer Express!



$2.95

31627 Dust Cover and Catalog

USE SINGLE SHEETS IN YOUR TRACTOR FEED PRINTER WITH PAPER TRACTOR®

Paper Tractor carries ordinary sheet paper through your tractor–feed printer! Simply place your letterhead, checks, invoices, forms, even legal–size sheets into the durable flexible plastic carrier and feed into your adjustable tractor–drive printer. Works with any printer, carries any paper.

22084 — $11.95

From Your Friends At
TENEX Computer Express

We gladly accept mail orders!
P.O. Box 6578
South Bend, IN 46660

Questions? Call 219/259-7051

Ad A1A

SHIPPING CHARGES

ORDER AMOUNT	CHARGE
less than $20.00	$3.75
$20.00–$39.99	4.75
$40.00–$74.99	5.75
$75.00–$149.99	6.75
$150.00–$299.99	7.75
$300 & up	8.75

MasterCard — American Express — VISA

NO EXTRA FEE FOR CHARGES
NO HIDDEN EXTRAS
ORDER TOLL FREE 1-800-348-2778

Changing Volume. Here is where you can have the most fun. By gradually changing the volume of a held note, you can make it fade and swell. Using sharp changes, you can make repeating engine noises. For instance, the *Helicopter* program starts at 0 volume and rises quickly to 15, then falls again, over and over. If it happened faster, it would sound more like a motorboat; if it happened slower, it would sound more like a chugging train.

Combining Voices. Most of the sound effects here use more than one voice, usually combining noise with musical sounds. With *Helicopter*, for instance, while voices 1 and 2 are making the noisy "chopper" sound, voice 0 is being sustained at a low level, playing a deep humming sound. The combination makes it sound like you can hear both the chopping of the blades through the air and the thrumming of the engine. To make the 'copter seem more distant, you simply delete voice 0's tone.

ONE SOUND IS WORTH A THOUSAND WORDS

There's a limit, of course, to how much I can explain—sounds have to be heard to be understood. The example programs included with this column show many different approaches.

In every program, lines 900 to 990 are a setup subroutine that merely assigns register addresses to variables. The programs for the 64 use the following variables for the register addresses:

AD(0), AD(1), AD(2): attack/decay registers for voices 0, 1, and 2

SR*(n)*: sustain/release registers 0-2.

FL*(n)* and FH*(n)*: frequency or pitch registers, low byte and high byte. Usually only the high-byte register is used for these coarse pitch manipulations.

GT*(n)*: gate or voice control registers 0-2.

VL: volume control register.

The programs for the VIC use the following variables for register addresses:

VC*(n)*: voices 0 through 2. Voice 2 is the noise register.

VL: volume control register.

In every program, lines 100 through 190 are used merely to access the sound effect—by pressing SHIFT you cause the sound to be activated.

In every program, the actual sound effect is produced in lines 20 through 90. To understand what is going on, study those lines. The first few lines usually set up the sound envelope (with 64 programs) and initial pitch and volume values. Then the sound is gated on, and the rest of the routine alters pitch and volume to make the sound effect.

Some programs, like the *Gunshot* variations, *Sawing Wood,* and *Boink,* make only one brief sound effect. To repeat them, you press SHIFT over and over. The *Helicopter* program repeats until you press SHIFT to stop it. The *Seashore* program repeats with randomized differences until you press RUN/STOP-RESTORE.

Once you understand what each program is doing, try changing small things, like the number of repetitions in a delay loop or the STEP value in a loop that changes pitch or volume. By changing one value at a time and RUNning the program, you can hear how the sound is shaped by each element of the routine. It won't be long before you'll be developing your own sound effects from scratch.

COMBINING WITH YOUR GAME

Often it is just fine for your program to stop cold while it executes a sound-effect routine. The sound effect when the player-figure is "killed," for instance, can take as long as you want it to take—the action is halted anyway.

However, many effects should go on while the action continues. This gets much trickier. You have to integrate the pitch and volume changes into your movement routines. Each POKE to change a sound will slow down the movement. As long as you're working in BASIC, that's the price you pay for sound, and you'll undoubtedly end up simplifying sound or movement or both in order to get adequate speed.

Professional games are written to use machine language routines running during interrupts and timed with system timers to create sound effects and play background music without slowing down the program. This is not possible in Commodore BASIC (though some BASICs, like the IBM PC's BASIC, have a background music mode that does pretty well).

But don't let that discourage you. Sound doesn't have to be continuous to be effective. If you have excellent music during lulls and some sound effects where they're most needed, the player will get the feeling of a complete game world.

And there's no law that says you can't devise sound effects even for word games and text adventures. Such games play slowly enough that a few clever sound effects won't noticeably slow down the action—and will add immeasurably to the fun of playing. Why not have the sound of somebody walking up or down a flight of stairs in a text adventure? Or the swish of a sword-slash that missed? Or the crash or an explosion? Why not have musical reminders that time is running out, fuel is running low, or a prisoner is running away?

NOTES ON PAST COLUMNS

For those of you who typed in the *Mansion Mystery* a few months ago and think I cheated you, here are some of the problems and how to fix them.

1. You ordered the disk and the program doesn't run. Source of the problem: a few disks were sent out with a trashed version of the program on it. Solution: send your disk back for a free replacement.

2. You get an error message when you first run the program. Source of the problem: if you ran the *Display Setup* program to create the disk or tape file of the mansion map, and then LOADed and ran *Mansion Game* without turning the machine off and on again, the *Mansion Game* program found the map already in memory and did not execute the setup routines. Solution: turn the machine off and on again, then LOAD and RUN *Mansion Game.* If you still get an error, then you have a typing mistake.

3. Weird things happen with the color when you first enter a new wing of the house. Source of the problem: you have a very early 64 which does weird things with color memory. Solution: buy a new 64 or live with the problem—I don't understand why it's happening on those few machines and haven't the faintest idea how to eliminate the problem. There's no logical reason it should be happening.

If you problem doesn't fall under any of these categories, then the program doesn't run because you have made a typing error. The unfortunate thing is that the program deliberately makes hash of screen memory, so that the error messages are often completely invisible. I suggest that you LOAD your copy of the program and check it against the *Bug Repellent*, or LIST it line by line, checking it against the printout in *Ahoy!* Or you can give up and send in for the disk...

That's the problem with mammoth programs like that. They're so long that typing them in is agony, finding typos is even harder, and then when it doesn't work you get homicidal thoughts. Please don't act out your fantasies of axe murder and slow torture on me. The disks I send in to *Ahoy!* always contain working versions of the programs—I LOAD and RUN every program just before mailing the disk. If you want to cause bodily injury to someone, I suggest you look at the masthead in this issue of *Ahoy!* and pick a name at random. The overworked, underpaid staff members at *Ahoy!* are always looking for interludes of violence and terror during the long, dull working day. □

SEE PROGRAM LISTINGS ON PAGE 91

SOUND-A-RAMA

A SOUND EFFECTS SHOWCASE FOR THE C-64

By Bruce Bartlett

Sound-A-Rama displays the incredible sound capabilities of the Commodore 64. This menu-driven program includes fifteen fantastic sound effects, a *Drum Machine*, and a *Computer Tape Recorder*. You type in the letter corresponding to the desired sound effect, and the effect plays. The letter of your choice lights up to indicate which effect is playing; otherwise the letters are light blue.

The *Drum Machine* is a subroutine containing six sounds of a drum set. You play it by hitting number keys 1 through 6. With the *Computer Tape Recorder*, you play music on the number keys. The computer "records" what you play, then reproduces the melody just as you played it—with correct pitches and timing.

For best sound quality (especially with the *Drum Machine*), you may want to play the computer sounds through a hi-fi system. To do this, you can obtain a long "patch cord" with an RCA phono plug on each end (available at Radio Shack). Plug your computer audio output into an "aux" input jack on the hi-fi system.

These sound effects can be added to your own programs by typing in the appropriate subroutines and initializing statements. Each effect is preceded by its title in a REM statement. Lines 1530-1550 should also be included; they clear the sound chip and define the sound parameters. In addition, the *Computer Tape Recorder* and *Drum Machine* have initializing statements at line 1860 and lines 1920-1930 respectively. The main loops for these subroutines are near the beginning of the program.

HOW THE PROGRAM WORKS

In the sound subroutines, up to three kinds of FOR-NEXT loops are used. FOR F = ... NEXT F increments or decrements frequency. That is, it sweeps the pitch from low to high, or vice versa. FOR D = ... NEXT D increments or decrements duration. It varies the duration or length of tones. FOR N = ... NEXT N increments the number of repetitions, to repeat an effect several times.

Described below are the definitions of the program variables. All the following variables refer to Voice 1 except as noted:

S = 54272, the start of sound-chip memory
FL = S = lower frequency (least significant bit)
FU = S+1 = upper frequency (most significant bit)
W = S+4 = waveform
AD = S+5 = attack/decay
SR = S+6 = sustain/release
W(2) = S+11 = Voice 2 waveform
FL(2) = S+7 = Voice 2 lower frequency
FU(2) = S+8 = Voice 2 upper frequency
AD(2) = S+12 = Voice 2 attack/decay
SR(2) = S+13 = Voice 2 sustain/release
SL = S+14 = Voice 3 lower frequency, used in ring modulation and synchronization modes.
SU = S+15 = Voice 3 upper frequency, used in ring modulation and synchronization modes.
S+22 = filter upper cutoff frequency
S+23 = filter and resonance modes
V = S+24 = volume and bandpass
F = frequency (when used in a FOR-NEXT loop)
D = duration (when used in a FOR-NEXT loop)
N = number of repetitions (when used in a FOR-NEXT loop)
POKEW,19 enables synchronization
POKEW,21 enables ring modulation

Described below are the functions of the program lines. Let's start at the program initialization:

Line 1530 clears the sound chip
Lines 1540-1550 define the sound parameters.
Lines 1560-1750 are the screen display, the main menu.
Line 1760 resets the attack/decay and sustain/release to nominal values for most of the sound effects.
Line 1770 restores a yellow letter to light blue after a sound effect has played.
Line 1780 gets keyboard input.
Line 1790 converts the typed-in letter to a number for use with the ON-GOTO statement in line 1830.
Line 1800 colors the typed-in letter yellow on the screen to indicate which sound effect is playing.
Lines 1810-1830 branch to the appropriate subroutines for each sound effect.
Lines 1860-1900 are the *Computer Tape Recorder* initialization.
Lines 1920-1970 are the *Drum Machine* initialization. (Note: initializations are put at the end of the program so that the main loops can go at the beginning of the program. This increases execution speed. For the same reason, numeric variables are defined as alphabetic variables in the *Computer Tape Recorder* and *Drum Machine* subroutines.

Lines 130-1500 are the sound effect subroutines.

Let's briefly describe what's happening in each sound-effect subroutine. I won't mention the ADSR settings since they are self-explanatory. In the last line of each effect, the voice is turned off. If the filter mode or resonance have been used, they are also turned off. Then the program goes back to line 1760 to reset the selected letter to blue and get a new keystroke.

Computer Tape Recorder is a program in itself. To play a note, the user hits a number key for a particular duration. The program (1) assigns the note's number to a number-array element, (2) converts the number to a frequency, (3) determines the note's duration (by counting elapsed time from TI$="000000"), and (4) assigns the duration to a duration-array element. This process is repeated each time a note is struck.

During playback, each successive element of the stored arrays is used to reconstruct the entered melody. That is, each number in the number-array is converted to a frequency, and is played for the duration indicated by each element of the duration-array. This process is repeated for each note in the melody.

Let's explain each line of *Computer Tape Recorder.* In this explanation, alphabetic variables have been converted back into numbers for clarity.

Line 130 gets keyboard input, and assigns each keystroke to an element of the number-array C$(N).
Line 160 can be rewritten as

```
160 POKES+4,16:D(N-1)=TI:TI$="000000":POKES+4,17:
POKES+1,10*VAL(C$(N) )
```

Line 160 turns off the waveform (POKES+4,16), assigns the duration TI of the note just played to a duration array D(N−1), resets the timer (TI$="000000"), turns on the triangle waveform (POKES+4,17), and plays the frequency 10*VAL(C$(N)).
Line 170 increments the array number (N=N+1) then returns to line 130 to get more notes.
Line 140 says, "If keyboard input is the letter 'P' and N>1, then play the recording." M is set equal to N, the index number of the last array element. Then N is set to 1 to reset the array to the beginning, and the program goes to Line 180.
Line 180 turns on the triangle waveform, plays the frequency indicated by the number-array element for the duration indicated by the duration-array element, then turns off the waveform.
Line 190 increments the array-element index. If the array-element index equals M (the index of the final element played), then the *Computer Tape Recorder* subrouutine reruns.

Drum Machine is another program-within-a-program. Line 220 gets keyboard input. Line 240 branches to the appropriate subroutine for each drum sound.

Each drum sound is a tone or noise set to a certain frequency and attack/decay/sustain/release. At the beginning of the "tom-tom" and "kick drum" sounds, the volume is quickly dropped and restored to maximum. This creates a "click" which simulates the drumstick hitting the drum.

Winding Down Record is a sequence of four notes played over and over with increasing duration and decreasing frequency.

Pulverizer enables synchronization, sets the Voice-3 upper frequency, and sweeps Voice 1 downward in pitch at a rate (STEP-N) that increases with each repetition. The sweep is repeated 20 times.

Disintegrator sets the Voice-3 frequency, enables ring modulation, and repetitively sweeps Voice 1 up in frequency at an ever-increasing rate, three times.

Old Washing Machine plays a triangle wave (POKE W,17) for a short duration, and turns it off with a sawtooth-wave shutoff (POKE W,32). This creates a noise burst. The frequency is doubled, and the process is repeated until the frequency equals 64. This entire process is repeated 20 times.

Energy levels sets the Voice-3 frequency, sets the Voice-1 AD/SR, enables ring modulation, and sets the Voice-1 upper frequency to a frequency that depends on time T.

Nervous sets the Voice-1 frequency, sets the Voice-3 upper frequency, enables ring modulation, and sweeps the Voice-3 lower frequency rapidly up and down 20 times.

Sproing first plays a short-duration low-frequency sawtooth wave. Then it sets the Voice-1 frequency, enables synchronization, and rapidly sweeps the Voice-3 upper frequency upward at an increasing rate, 50 times.

Force Field sets the Voice-3 frequency, sets a slow attack, sets the Voice-1 frequency, enables ring modulation for about five seconds, then fades out the volume.

Involution enables the bandpass filter, sets maximum resonance, turns on the triangle waveform, and repetitively sweeps down the frequency and the bandpass center frequency at a gradually increasing rate.

Aviary sweeps upward between two random frequencies 15 times, then sweeps upward rapidly (warbles) a random number of times. This process is repeated.

Ghost sweeps the Voice-1 frequency rapidly up and down, while the average frequency gradually decreases. This process is repeated.

Song of Saturn turns on maximum resonance, turns on the bandpass filter, plays noise at a random frequency, and sweeps the bandpass-filter center frequency from 0 to a random number. This process is repeated 10 times.

Ketchup Squirts turns on maximum resonance, turns on the bandpass filter, and sweeps the bandpass filter upward at a random rate while playing a random-frequency sawtooth wave.

Elf Laugh plays a sawtooth-wave tone burst followed by a noise burst (POKE W,128), at successively lower frequencies, 10 times. This forms a "laugh cycle." The average frequency is raised each time the laugh cycle is repeated (three times).

Throb sets the Voice-1 upper frequency. Then, while sweeping the Voice-1 lower frequency down slowly, it plays a sawtooth-wave tone burst followed by a triangle-wave tone burst. □

SEE PROGRAM LISTING ON PAGE 94

OPENING CEREMONIES

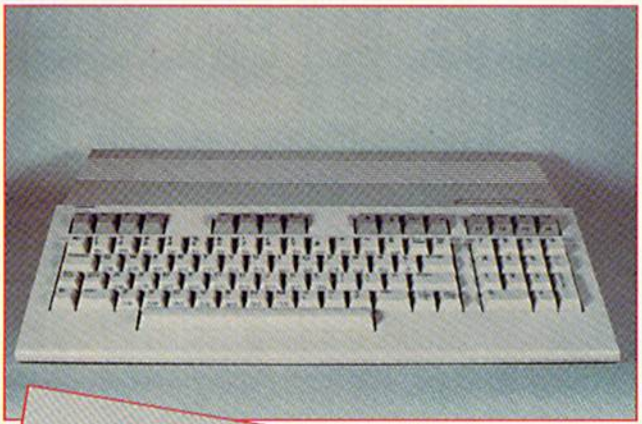
for the Commodore 128

By
Morton Kevelson

The arrival of the C-128 at our offices on April 10th heralded Commodore's leavetaking from the home computer business. As the logo proclaims in no uncertain terms, the C-128 is a personal computer. Personality is definitely one thing the C-128 has plenty of. According to some counts it can be credited with no fewer than five.

Many of us will be intimately familiar with one of its personalities, that of the C-64. As promised, there is a full-fledged C-64 hidden within the confines of the C-128. There are three ways to bring out the C-64. For starters, simply hold down the Commodore logo key when you power up the computer. This will automatically bring forth the C-64 within. If a C-64 cartridge is left in the expansion port, the C-64 mode will automatically be selected. The Epyx *Fast Load* cartridge serves admirably well in this respect. From C-128 mode you can tell the machine to GO 64. However, you will have to be firm. The C-128 will promptly question this demeaning request. Just respond in the affirmative and the familiar blue on blue of the C-64 display will blossom forth on your video monitor.

A true C-64 it surely is. Commodore has gone to great lengths to insure compatibility with all existing C-64 software and peripherals. Even the nice numeric keypad located at the extreme right of the keyboard (see photo) is disconnected in C-64 mode. Once in C-64 mode, the computer resists all attempts to leave it. Even turning it off does not always work. We found that it took more than a few seconds in the benign state for all feelings of C-64ishness to dissipate. But enough of this talk about the old. Let's get on with the new.





Top: all 92 keys of the C-128. Main section nearly duplicates the C-64. Bottom: closeup showing numeric keypad.
READER SERVICE NO. 175

THE KEYBOARD

This is definitely a step in the right direction. The 92 keys have a nice feel to them. Touch typists will be pleased to find a raised dimple on the home keys (F and J). Accountants will likewise appreciate the similar treatment of the 5 key in the numeric keypad. This keypad should go a long way toward easing some of the tedium associated with marathon spreadsheeting into the wee hours. We would have liked to have seen the parenthesis, asterisk, and slash in the keypad as well. However, we admit that these are the least likely to be used for most accounting applications.

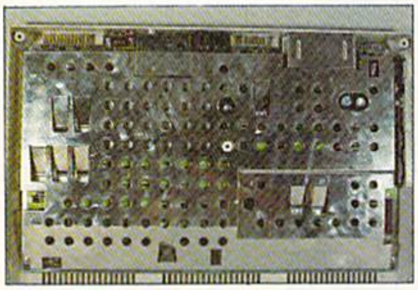
With the exception of the function keys, all the keys on the top row are new. These are not just for show. The ESCape key supports a host of screen editing functions. These include erase to end of screen, erase to end of line, erase to beginning of line, automatic insert mode, insert line, delete line, scroll screen, set window, and toggle between 40/80 column display. All of these sequences work under program control. To use them simply PRINT CHR$(27), followed by the appropriate command letter.

Incidentally, the escape character can be generated by holding down the CONTROL key and pressing the colon (:). This is true for both the C-64 and the C-128. The results will be different for the two machines. The

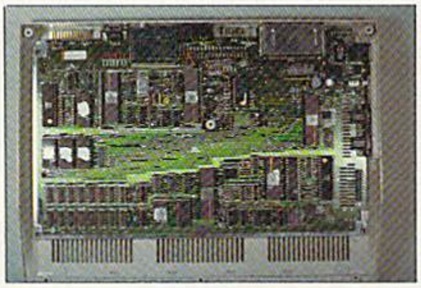
C-128 is always placed into ESCape mode awaiting a command character for immediate execution. The C-64 will generate a reverse-left-square-bracket when in quote mode. Thus, for the C-64 the above key sequence can be used to put an ESCape code into a PRINT statement.

Some very subtle alterations are buried beneath the keyboard. Turn up the volume on your TV or monitor, hold down the CONTROL key, and press G. A pleasing bell tone will sound. PRINT CHR$(7) will elicit the same effect. This keyboard "bell" is a standard ASCII code from the early days of teletype terminals; a useful feature which was sorely missed on the VIC and C-64.

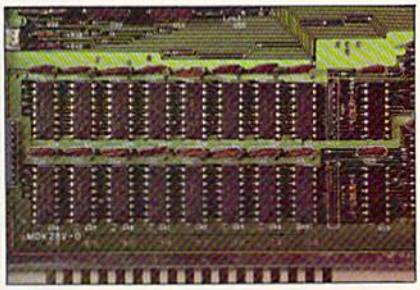
The ALT key is to access an alternate character set under program control. The HELP key is for use immediately after a SYNTAX ERROR?.



The upper radiation shield of the C-128 serves as a heat sink as well.



The main circuit board sports a neat and uncluttered layout.



Sixteen 64 kilobit dynamic RAM chips make up the 128K of main memory.

The offending BASIC line is displayed with the part following the error in reverse. The 40/80 DISPLAY key determines the screen display mode on power up. It has no effect once the computer is turned on unless the reset button is pressed.

BASIC 7.0

This version of Commodore BASIC is definitely loaded. Gone are the days of POKEing and PEEKing to generate sounds or program sprites, or manipulate the color display. There are now specific BASIC commands to handle all of these chores. Included are a built-in sprite editor and music editor. The accompanying table summarizes all BASIC 7.0's enhancements as compared to 2.0.

Acoustically inclined programmers will be particularly pleased with the ENVELOPE and SOUND commands. The former includes ten built-in musical instruments. The latter supports the creation of complex special effects. For musicians, the PLAY command accepts musical notation. In fact, the built-in sound commands have more features than some music editor programs we have seen.

The programmer's aid commands will ease many of the more tedious programming tasks. The function keys are preprogrammed with useful phrases. The KEY command easily reprograms the function keys for even greater profundity. Lines can be AUTOmatically numbered, RENUMERED and DELETEd en masse. The proper effects are extended to GOTOs and GOSUBs. The TRAP, trace (TRON, TROFF), RESUME, and HELP commands should keep rampant SYNTAX ERRORism under control. About the only significant shortcoming is the lack of a string FIND or SEARCH command in the programmer's aid section.

The structured programming commands deserve special attention, in particular the IF..THEN BEGIN: (program lines):BEND:ELSE sequence. The Block-BEGIN and Block-END structure allows the writing of complex routines as part of an IF..THEN decision without the need of GOTO or GOSUB. This should definitely promote much neater and easier to follow program structures.

Numerous other improvements abound, not the least of which is memory. Even the most determined BASIC programmer will have a tough time exhausting the available supply. The startup message proclaims a massive 122365 bytes free. This is nearly 120 kilobytes as computers count. There is one catch. BASIC memory is partitioned into two banks. Bank 0, which is used for the program storage, has 58,109 bytes available. Bank 1 provides 64,256 bytes for variable storage. Note that memory usage is restricted to these predefined applications.

The full screen editor can accommodate program lines up to 160 characters long. It links four screen lines when in 40 column mode and two screen lines in 80 column mode. This is certain to result in competitions for the longest LISTing program line amongst aficionados of compact code.

Long garbage collection delays should also become a thing of the past. Try the following in both C-64 mode and C-128 mode:

```
10 DIM A$(255)
20 FOR J=1 TO 255
30 A$(J)="A"+"B"
40 NEXT J
50 T=TI
60 M=FRE(1)
70 PRINT TI-T
```

The number displayed is the time in jiffies (1/60 second) it took to perform garbage collection. Try it in both C-64 and C-128 modes. For the C-64 it should be about six seconds. For the C-128 it will be about 1/6 of a second!

THE DISPLAYS

Both 40 and 80 column displays are built in. In 40 column mode the default colors are light green on a dark grey background with a light green border. For 80 column mode

C-64 and C-128* OWNERS:



Now YOU too can get the "Software Edge!"

Do you know the full contents of all your Disks in your Software Library?

Stop playing "Hide-and-Seek" with your Disk software! Let your Commodore 64™ "memory" do all the work for you. Easy Manual and Program Menu-Screens make everyone a "pro" in 1 day!

Here's a small "PEEK" at SOME of the FULLY-INTEGRATED POWER of PRODUCTIVITY at your fingertips. . .

MASTERDISK©:

- a SORTED Master Catalog of ALL your Disks (even those un-listable "protected" Directories: Names, ID's, File info, etc.)
- Search + List any File/Disk/ID in seconds, automatically, in any of SEVEN WAYS.
- Transfer ML-Programs or Files from Disk to Disk (CATEGORIZE your software)
- Stop disaster before it strikes: change duplicate ID's fast, without total re-copy.
- Restore accidentally-scratched files in a flash, with VERIFICATION that they're OK.
- Works with 1 or 2 Single Disk Drives (1541, MSD, etc.) with or without Printer (Commodore or ASCII).
- Handy-Reference Card for HEX/ASCII/ BINARY-TO-DECIMAL VALUE TRANSLATION + FORMULAS, included FREE.
- All this + MUCH MORE, for just $34.95. (U.S.A.), $45.95 (Canada)

NEW!! - MASTERDUAL©, "Son of MASTERDISK©". . . same power, but with Dual Drive Speed!

- Same Catalog Features as MASTERDISK©
- Same Disk repair functions, too
- For all C-64 & C-128* systems with Dual Disk Drives (4040, MSD, etc.)
- Utilizes all Dual Drive functions for maximum speed and ease of use
- FREE Handy Reference Card for HEX/ASCII/ Binary to Decimal Translation + Formulas
- Experience FULLY-INTEGRATED POWER for just $29.95 (U.S.A.), $39.95 (Canada)

*MASTERDISK© and MASTERDUAL© are also compatible with the NEW C-128 (in C-64 Mode)!

Don't MESS with anything LESS! ***OUR CUSTOMERS WRITE:***

"Been meaning to write and let you know how much I've enjoyed it . . . So, let me say that MASTERDISK is EXCELLENT." — Middletown, NY

". . . just the thing I need." — APO, NY (Germany)

"Hope to get my [MASTERDISK] yesterday." — Maple Grove, MN

"I am impressed with your product and would like to be able to endorse it for TPUG." — Ontario, Canada

". . . nothing else even comes close!" — Burlington, IA

We ship within 24 hours! Send Money Order or Check to:

INTEGRATED-SOFTWARE SYSTEMS
P.O. Box 1801, Ames, IA 50010
VISA/M.C./C.O.D., CALL **(515) 233-2992**
9 am - 9 pm (Central Time) Mon.-Fri.
Add $3 Shipping/plus $2 Foreign, C.O.D.
IA Residents add 4% Sales Tax.

IN CANADA: Cricket Distribution Co.,
P.O. Box 953, Cobourg, Ontario, Canada K9A 4W4
VISA/M.C./C.O.D., CALL **(416) 372-3692**
9 am - 6 pm (Eastern Time) Mon.-Fri.
Add $3 Shipping/plus $2 C.O.D.
Ontario Residents add 7% Sales Tax

Whether you have tens or hundreds of disks, now YOU can be the MASTER of your TOTAL Software Library!

you will need a high resolution RGB monitor for best results. In a pinch the 1702 monitor front connection could be used (see sidebar on this page). The results were far from ideal, without color, but better than the soft 80 column displays we have seen on the C-64. We did discover that the character matrix in 80 column mode is only six pixels wide. Traditional Commodore characters are all eight pixels wide.

Both displays remain active at all times. There is even a dedicated two kilobyte static RAM chip on board for the 80 column display. Thus you can write messages on the 40 column screen while working on the 80 column display. Of course two monitors are required for maximum benefit. Since the display mode is switchable under program control, this could become an added bonus for many applications. Note that only the 40 column mode is supported by the BASIC 7.0 graphic commands. According to Commodore's Jim Gracely, the C-128 will have some bit map capability in 80 column mode. The horizontal resolution will be a full 620 pixels.

The 80 column display has several features not available in the 40 column mode. Onscreen underlining can be turned on with a CONTROL B (CHR$(2)) and turned off with a CHR$(130). The underline feature also "eats" the pixel which displays the lower case descenders of the screen characters. Characters can be set to flash at the cursor blink rate with a CONTROL O (CHR$(15)). A CHR$(143) turns flash off. Although neither a CHR$(130) nor a CHR$(143) can be generated directly from the keyboard, both flash and underline will be turned off by a carriage return.

Commodore plans to market two monitors for the C-128. The 1901 will be a monochrome monitor suitable for either 40 or 80 column work. The 1902 will be a dual purpose monitor with both RGBI and composite video inputs. A switch on the front panel will select its operating mode. This will allow both C-128 displays to be easily accessed by a single monitor. All that is lacking is the ability for the C-128 to switch the monitor's mode under program control. If you are buying a non-Commodore monitor, bear in mind the intensity requirement of the RGBI input. Some digital type monitors lack this analog capability. These may not be suitable for the C-128. The RGBI port on the C-128 is similar to that on the IBM PC. Thus monitors which work with that machine should work with

Displaying 80 Columns with the C-128 and 1702 Monitor

Although a high resolution color monitor with RGBI (red, green, blue, and intensity) inputs is needed for the best results, it is possible to get a usable 80 column display with the C-128 and Commodore's 1702 color monitor. This will allow the use of 80 columns, with a minimum of investment, while waiting for Commodore to distribute the new 1902 color monitors.

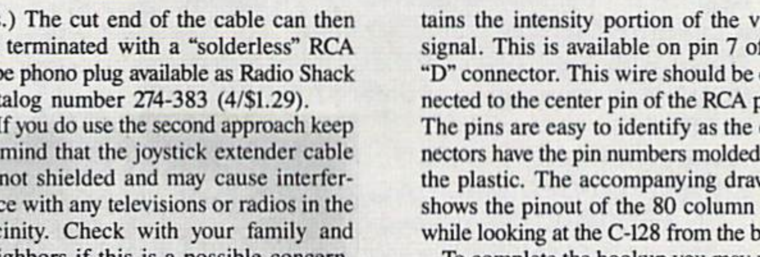
All that is needed is a suitable adapter cable. The 80 column video port on the C-128 uses a readily available 9 pin "D" type connector. This is the same type of connector used for the joystick ports on the VIC 20, C-64, and C-128. It is available as Radio Shack catalog number 276-1537 for $1.99. If you use this connector you will have to be able to solder on a length of shielded coaxial cable. Audio type cable works fine for the short length which is needed. The other end of the cable will be terminated with a standard RCA type audio plug.

Soldering can be avoided by using a joystick port extender cable. These are available for under $10.00 at most stores which sell joysticks and other video game accessories. Just cut the cable in two, making sure that a sufficient length of wire remains attached to the "male" connector end of the cable. That is the one with the pins. The "female" end has the sockets for the pins. (We are not certain how this chauvinistic terminology came about but we understand that it dates back to the early days of radio and electronics.) The cut end of the cable can then be terminated with a "solderless" RCA type phono plug available as Radio Shack catalog number 274-383 (4/$1.29).

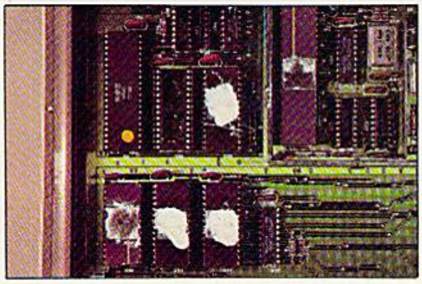
If you do use the second approach keep in mind that the joystick extender cable is not shielded and may cause interference with any televisions or radios in the vicinity. Check with your family and neighbors if this is a possible concern. You will also need access to an ohmeter or other continuity checker to determine which of the nine wires in the cable go to the proper pins. Also be sure to insulate the seven unused wires to prevent any electrical shorts.

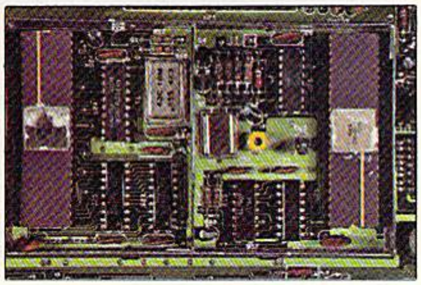
Only two wires have to be connected. One of these is the ground connection available on both pins 1 and 2 of the "D" connector. Either one will do. This wire should be connected to the shield or outer part of the RCA plug. The second contains the intensity portion of the video signal. This is available on pin 7 of the "D" connector. This wire should be connected to the center pin of the RCA plug. The pins are easy to identify as the connectors have the pin numbers molded into the plastic. The accompanying drawing shows the pinout of the 80 column port while looking at the C-128 from the back.

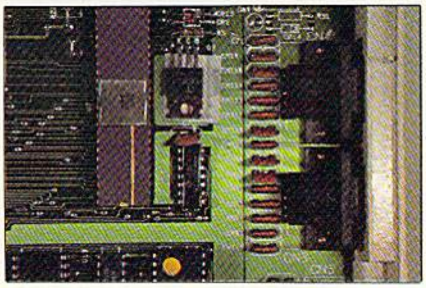
To complete the hookup you may want to insert a "Y" audio adapter cable into the audio line at the back of the monitor. This will make available the sound portion of the signal from the 40 column port to both the front and back connectors. This will not be needed if you normally use the front connection on the 1702 monitor. Just leave the audio plug where it normally is. If you are using the back connection for 40 column display (the best arrangement) then the selector switch on the back panel of the 1702 can



Pinout of 80 column video port as seen from rear of the C-128. Pin 1 or 2 is used for ground connection, pin 7 for the intensity portion of the video signal.







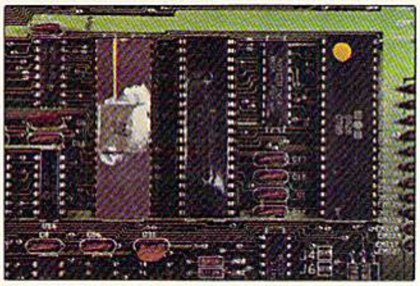
Left: empty 28 pin socket between 16K ROM and 6526 CIA chip. 48K of ROM can be seen here. Center: dual video display sections allow simultaneous 40 and 80 column displays. Right: twin joystick ports.

the C-128 as well.

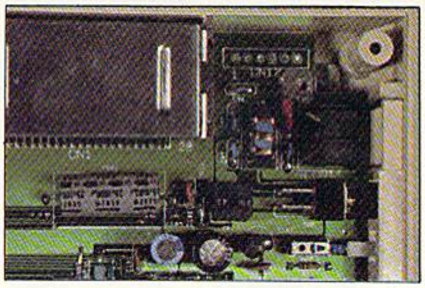
THE HARDWARE

As with all recent Commodore computers, the C-128 comes in two parts, a power supply module and a keyboard/CPU. The power supply is massive, with more than twice the volume of the C-64 power supply. It also has a replaceable fuse and a screw-together case. Unlike the C-64, this power supply can be repaired. We don't know what Commodore has in mind, but with just the C-128 online the power supply barely got warm. We do hope that Commodore will continue to supply this open construction, as opposed to the potted design currently in use for the C-64.

Inside the computer, the main circuit board is fully shielded. The upper shield also serves as a heat sink for the large scale integrated circuits.



Left: 8502 microprocessor for 64 and 128 mode; right: Z-80 for CP/M.



Clockwise from top: cart port, power connector, on/off switch, reset button.

A series of springlike "fingers," stamped in the sheet metal, reach down and touch the chips in the appropriate spot. A dab of thermally conductive silicon compound ensures good heat transfer. We found no significant hot spots after extended periods of operation.

A minimum of 48 kilobytes of ROM make up the C-128 operating system and BASIC. Keep in mind that 16 kilobytes are actually the original C-64 BASIC and Kernal. This still leaves twice that amount for the enhanced features of the C-128. In the midst of the ROM cluster we found an empty 28 pin socket. This opens the possibility of future ROM expansion, perhaps some built-in application software.

The accompanying photographs tell the entire story.

EXPANSION RAM

Commodore plans to make up to 512 kilobytes of expansion RAM available for the 128. It has not been decided as of this time whether this will be available only as a RAM disk option or as bank-switched RAM. The present thinking is some kind of combination of the two. Details will follow when available.

WHAT'S NEXT?

We still haven't touched on the built-in CP/M* capability. According to the accompanying manual, the CP/M 3.0 Plus operating system will be able to access all of the C-128 color and sound capabilities as well as both the 40 and 80 column displays. When using the 40 column display the screen becomes a horizontally scrolling window on an 80 column line. Although not ideal, this does present a workable compromise to allow the use of 80 column CP/M software with a 40 column display. The

be used to switch between the 40 and 80 column displays. Of course computer output will still have to be directed as required.

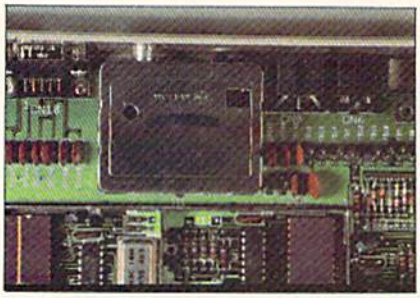
To finish off, adjust the brightness and contrast controls on the monitor front panel for best results. We found that they had to be turned down quite a bit to obtain a usable image. Color in 80 columns will not be available with this connection. However, the flash and underline features will work.

We haven't tried this with a 1701 monitor but we see no reason for it not to work with that device as well.

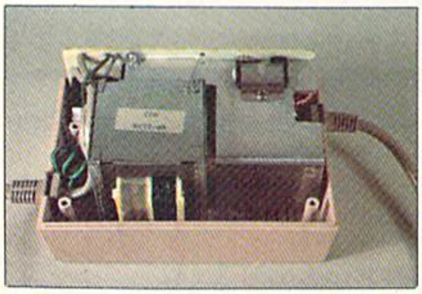
An even sharper picture will result if the 80 column screen is reversed to a white background by an ESCape R command sequence.

80 COLUMN RGBI PORT PINOUT

Pin #	Description
1	Ground
2	Ground
3	Red signal
4	Green signal
5	Blue signal
6	Not used
7	Intensity signal
8	Horizontal drive
9	Vertical drive



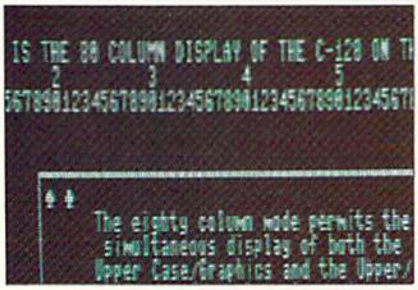
Top, l to r: RGBI connector, RF modulator, composite video and serial ports.



Power supply: transformer (left), rectifier, and voltage regulator boards.



An 80-column display using the 1702 color monitor (see sidebar, p. 32).



Closeup of 80 column display on 1702.

CP/M system disk and a utility disk will be supplied with the computer. Two more CP/M utility disks will be supplied with the 1571 disk drive.

As of this writing we are still waiting for the CP/M system disk, required (along with the 1571 disk drive) to book CP/M on the C-128. Interestingly enough, of all the C-64 accessories we tried, only the original C-64 CP/M cartridge would not work. Of course this can be construed as true compatibility, as we know of many C-64's which will not work with the cartridge. □

BASIC 7.0 EXTENSIONS TO BASIC 2.0

The following commands are included in BASIC 7.0 in addition to all of the BASIC 2.0 commands.

Programmer's Aid

AUTO [increment]—automatic line numbering with specified increment
DELETE [line#][—line#]—deletes specified line or range of BASIC lines (direct mode only)
HELP—displays line with BASIC program error
KEY [key#,string]—assigns string to specified function key
MONITOR—calls the built-in machine language monitor
RENUMBER [new starting line# [,increment[,old starting line#]]]—renumber BASIC program (direct mode only)
RESUME [line#/NEXT]—continues program after TRAPping an error
TRAP [line#]—intercepts BASIC program errors, branches to specified line#
TRON—trace mode, displays line# of currently executing BASIC statement
TROFF—turns off trace mode

DOS Support

The disk drive command channel (secondary address 15) is automatically OPENed by these commands as required.

APPEND#lf,fname[,Ddv#][ON Udv#]—opens the specified file for writing. Appends subsequent data to end of file
BACKUP Ddr# TO Ddr# [,ON Udv#]—duplicate command for dual drives
BLOAD "fname"[ON][,Bbank#] [,Pstart addr]—performs a binary load to the specified bank and address
BOOT "fname"[,Ddr#][,Udv#]—loads a binary file and executes at the start address
BSAVE "fname"[,Bbank#] [,Pstart addr][TO Pend addr]—saves a binary file
CATALOG—see DIRECTORY
COLLECT [Ddr#][,ON Udv#]—same as BASIC 2.0's VALIDATE command
CONCAT [Ddr#,]"source file" TO [Ddr#]"dest. file" [ON Udv#]—merges two files
COPY [Ddr#,] "source file" TO [Ddr#,] "dest. file" [ON Udv#]—same as BASIC 2.0
DCLEAR Ddr#[ON Udv#]—clears all open channels
DCLOSE[#lf][,ON U dv#]—closes all or a single disk file
DIRECTORY [Ddr#][,Udv#] [,"filename"]—displays disk directory without affecting current program
DLOAD "name"[,Ddr#][,Udv#]—LOADs from disk
DOPEN#lf,"fname"[,Ly] [,Ddr#][,ON Udv#][,W]—opens a sequential (relative if L is present) file for read or write
DSAVE "name"[,Ddr#][,Udv#]—SAVEs to disk
DVERIFY "fname"[,Ddr#] [,Udv#] [,relocate flag]—verifies a disk file against memory
HEADER "name"[,Iid#], Ddr#[,ON Udv#]—formats a disk; BASIC 2.0's NEW command
RECORD#lf,rec#[,byte#]—adjust relative file pointers
RENAME [,Ddr#]"old name" TO "new name"[,Udv#]—renames a disk file
RUN "fname"[,Ddr#][Udv#]—LOADs and RUNs the specified file
SCRATCH "file name" [,Ddr#] [,Udv#]—erases a disk file

Notes for DOS commands

dr# = drive number, i.e. 0 or 1
dv# = device number, i.e. 8 or 9
id# = two character disk ID
lf = logical file number

Structured Programming

BEGIN/BEND—defines a block of code for use after THEN and ELSE
DO [UNTIL bool arg WHILE bool arg] program statements [EXIT] LOOP [UNTIL bool arg WHILE bool arg]
IF bool arg THEN statement [:ELSE statement]
bool arg = logical expression which is either true or false (−1 or 0)

Enhanced User Interface

BANK—selects 64K RAM bank for

***CP/M is a registered trademark of Digital Research Corporation.**

PEEK and POKE
GETKEY–same as 10 GET A$:IF A$= ""THEN GOTO10
GO64–switches to Commodore 64 mode
PRINT[#lfn] USING format list;print list;–formatted PRINT statement
PUDEF"1 to 4 characters"–redefines PRINT USING symbols
RREG [a var][,[x var][,[y var][,s var]]]–assigns microprocessor register values to specified variables on return from a SYS call
SLEEP N–delays program for N seconds

Graphics Commands

BOX [c-s#],x1,y1[,x2,y2][,angle][,paint]]–draws a rectangle
CHAR [c-s#],col,row[,text][,rvs-flag]–displays text on both the text and the graphic screens; equivalent to PRINT AT
CIRCLE [c-s#],[xc,yc],xr[,yr][,[sa][,ea][,angle][,inc]]]]]–draws a circle
COLLISION type[,line#]–defines handling of sprite/background/light pen collision interrupt
COLOR c-s#,color#,–assigns colors to one of the seven color sources
DRAW c-s#[,x1,y1[TOx2,y2...]]–draws dots, lines, and shapes
GRAPHIC mode[,clear[,s]/CLR–allocates or deallocates a ten kilobyte graphic area for bit mapped graphic mode; selects between 40 and 80 column text; s is first line of text for split screen
LOCATE x,y–places the pixel cursor on the screen
MOVSPR #,x,y–moves or locates sprite # specifies sprite number; x and y can be absolute or relative positions or angle and speed
PAINT [c-s#][,[x,y][,fill-mode]]–fills an area with color
SCALE [1/0][,xmax,ymax]–scales drawing coordinates as specified; defaults 0 to 1023
SCNCLR–clears the current screen
SPRCOLOR–sets multicolor 1 and 2 for all sprites
SPRDEF–calls built-in sprite editor
SPRITE #[,[on/off][,[fgnd][,[priority][,hor. exp.][,[vert. exp][,mode]]]]]–sets various sprite attributes
SPRSAV–saves a sprite image
SSHAPE string-variable,x1,y1[,x2,y2]–saves a rectangular graphics screen as a string
GSHAPE string-variable[,[a,b][,put-mode]]–displays a saved shape
WIDTH–sets width of drawn lines in graphic mode
WINDOW–defines a screen window on the text screen

Notes for graphics commands

c-s# = color source number
0 = 40 column background
1 = graphic foreground
2 = graphic multicolor 1
3 = graphic multicolor 2
4 = 40 column border
5 = character color (40 or 80 column)
6 = 80 column background

mode = graphic display mode

0 = 40 column normal text screen
1 = high-resolution graphics screen
2 = high-resolution graphics, split screen
3 = multicolor graphics screen
4 = multicolor graphics, split screen
5 = 80 column text screen

col = text column (0-79) (wraps in 40 column mode)
row = text row (0-24)
xr, yr = radii for circle
sa = start angle
ea = end angle

Sound and Music Commands

ENVELOPE n,[atk][,[dec][,[sus][,[rel][,[wf][,[pw]]]]]]–defines musical instrument envelope; ten musical instruments (piano, accordion, calliope, drum, flute, guitar, harpsichord, organ, trumpet, xylophone) are predefined; all may be modified

n = envelope (instrument) number (0-9)
atk = attack rate (0-15)
dec = decay rate (0-15)
sus = sustain (0-15)
rel = release (0-15)
wf = waveform:
  0 = triangle
  1 = sawtooth
  2 = pulse
  3 = noise
  4 = ring modulation
pw = pulse width (0-4095)

FILTER [f][,[lp][,[bp][,[hp][,[res]]]]]–defines sound filter parameters

f = cut of frequency (0-2047)
flags; 1 = on, **0** = off:
  lp = low pass
  bp = band pass
  hp = high pass
res = resonance (0-15)

PLAY "Vn,On,Tn,Un,Xn,notes/elements" where VN = Voice (n=1–3)
On = Octave (n=1–6)
Tn = Instrument envelope (n=0–9)
Un = Volume (n=0–15)
Xn = Filter on/off
Notes: A,B,C,D,E,F,G
Elements:
  # sharp
  $ flat
  W whole note
  H half note
  Q quarter note
  I eighth note
  S sixteenth note
  . dotted note
  R rest
  M wait for all voices to end

SOUND v,f,d[,dr[,m[,s[,wf[,pw]]]]]–creates and outputs sound effects
  v = voice (1-3)
  f = frequency (0-65535)
  d = duration in sixtieths of a second
  dr = frequency sweep direction
    0 = increment
    1 = decrement
    2 = oscillate (up, down)
  m = sweep minimum frequency (0-65535)
  s = sweep step value (0-32767)
  wf = waveform (see ENVELOPE)
  pw = pulse width

TEMPO–define note duration

VOL–set sound level or volume

Reserved Variables

DS–reads current drive status from disk error channel
DS$–reads current drive error message
ER–last program error after a RUN
EL–line number for ER

Functions

BUMP (n)–returns sprite collision information
DEC(hexadecimal string)–converts hexadecimal to decimal
ERR$(n)–returns program error message
FRE (n)–number of available bytes in specified bank
HEX$(n)–converts decimal to hexadecimal
INSTR(string1,string2[start-position]–finds position of string 2 inside string1
JOY(n)–returns position of the joystick
PEN (n)–returns coordinates of light pen
POT (n)–returns paddle setting
RCLR(n)–returns color assignment
RDOT(n)–returns current position of the pixel cursor
RGR(X)–returns current graphic mode
RSPCOLOR (reg)–checks sprite multicolor values last set
RSPPOS–checks speed and position of sprite
RSPRITE–returns sprite attributes
RWINDOW–returns current screen window parameters
XOR–exclusive or function

NEXT MONTH: Morton Kevelson continues his series on the Commodore 128 and its peripherals with a text and photo treatment of the 1571 disk drive, including information on specifications, compatibility, internal workings, and DOS 3.0; plus, an illuminating look at (and under) the C-128 keyboard.

WIND

on the VIC

BY THOMAS

Here is a neat program that will enable you to add a window to your programs to give them a professional touch. This window, a rectangular portion of the screen framed with a solid border and containing any message you wish, can be placed anywhere on the screen and will appear to overlay what was on the screen when an "open window" command is issued. The message contained in the window can either be selected before the window is opened or the output from the keyboard can be sent directly to the window. When a "close window" command is issued the window will be restored to its original condition. Since the program is written in machine language these changes occur instantaneously. Some of the uses for this feature include menus, help screens, and warning messages, as well as the ability to save or move sections of the screen.

There are separate programs for the VIC and 64. Both consist of a machine language program in the form of data statements and a BASIC loader to put the program into the computer's memory. The 64 version loads into a 1230 byte block starting at memory location 49408. The VIC version loads into the top 418 bytes of the BASIC programming area with the cassette buffer and 95 bytes starting at memory location 673 used for temporary data storage.

PROGRAM DESCRIPTION

The program uses three window commands to activate the program and five variables which tell the program the size, location, and contents of the window.

The "open window" command first saves the character and color information in the area of the screen where the window is to be placed. This information is stored in a file contained within the window program. A window containing the desired message is then printed to the screen.

A "write window" command does the same thing, except it does not save the information on the screen, thus any information the program has previously stored in its file remains undisturbed. This command is used to change the contents of a window already on the screen.

A "close window" command replaces the window with the character and color information that was saved when the window was opened. This will restore the screen to its original condition.

Neither of these last two commands will disturb the screen information previously saved by an "open window" command.

Five variables are used by the program for window parameters, as follows:

W$ –window message
W% –window width
H% –window height
R% –row number
C% –column number

The first variable, W$, is the message the program will put into the window. If the string is not long enough to completely fill the window, spaces will be used after the string is exhausted. If W$ is undefined, the program will fill the window with garbage.

The next two variables, the width and height, can have any values that will not exceed the screen size, with one limitation: the product of the width times the height cannot exceed 95 for the VIC or 255 for the 64.

The last two variables set the position of the window on the screen. The row number, counting from the top, and the column number, counting from the left, specify the location of the upper left corner of the window (start counting with 0). Any values can be used that will not cause the window to run off the right side or bottom of the screen.

Any time a window command is issued the program first checks to insure that the values assigned to the variables meet the conditions outlined above. The program will abort and return to BASIC if an error is found.

The use of variables to provide instructions for the window program eliminates the need to POKE numbers into memory. They are created the same as any other variable used in your program and are stored in the same place. Every time the window program is activated the program will find the current values of the variables.

If you change the value of R% or C% before you issue a "close window" command, you will replace the screen section in a different spot on the screen, a feature that allows you to move a block of information from one part of the screen to another. If you change W% or H% before you issue a "close window" command, you will usually end up with a scrambled version of the original screen. Regardless, as long as the window size and location variables are changed back to the values they had when the window was "opened," a "close window" command will restore the information in the window area exactly as it was originally. Of course, if you have cleared the screen or allowed it to scroll in the meantime, the remainder of the screen will no longer match.

LOADING THE PROGRAM

After you have typed in the program, enter RUN (be



sure you have saved the program first) and a machine language program will be loaded into the computer's memory; the BASIC portion of the program will be destroyed and the commands you need to activate the window commands will be printed on the screen. For a 64 you will see:

Open window – SYS 49432
Write window – SYS 49436
Close window – SYS 49440

If you're using a VIC 20, different numbers will be displayed.

The procedure described above is the normal method for using the window program and uses the least amount of memory. It does have the disadvantage of requiring you to first load and run the window program, then load and run your program every time you want to use the window feature. If you have about 2K of RAM to spare you can avoid this two-step loading procedure as follows: load the window program but do not run it. Delete lines 6, 7, and 8 and proceed to write or append your program after the end of the window program. You will have to start your program line numbering with 35 or higher, but the window program including data statements will be an integral part of your program and will load and save with it. When you run the combined program the machine language program contained in the data statements will be loaded into memory, but the window commands will not be printed on the screen. You will not need to be concerned with the actual value of these memory locations, as you can make use of the variables that are defined in line 5 as follows:

Open window – SYS OW
Write window – SYS WW
Close window – SYS CW

WINDOW MESSAGES

When you compose a message to be put into a window there are several restrictions you should be aware of. Only letters or symbols can be used – no special function keys such as CRSR or CTRL keys. Because of the limited ASCII to screen code conversion routine, any time you press any key (except those on the top row of the keyboard) while holding down the SHIFT key you will get the character you expect, but it will be in reverse mode when printed in the window. The result will be a reverse mode upper case letter when in the text mode or a reverse mode graphics symbol when in the graphics mode.

Writing a string for use in a window can be a little tricky at first. After allowing for the border the usable width of a window is two less than the specified width. You could add spaces to pad out a line to achieve the desired format, but there is an easier way. The program has a feature to avoid the need for counting spaces. Whenever you want to start a new line in the window insert a £ in the string. The £ itself will not appear, but functions much as a RETURN key does when writing on the whole screen. The following character will appear at the left side of the window on the next line down.

You cannot change the character color within a window. If you want the window and its contents to be a different color, change color before opening a window and change back afterwards.

USING THE PROGRAMS

After you have loaded and run the program and obtained the screen display listing the window commands, try the following to become familiar with the operation of the program. In the examples that follow the numbers that follow the SYS commands are for a 64. A VIC will require substitution of the numbers printed on the screen when the program is run. Enter in direct mode:

```
W%=8: H%=9: R%=0: C%=12: W$="HELLO": SYS
49432
```

This should print a window 8 spaces wide by 9 spaces high containing the word HELLO. The left corner will be in the top row and moved 12 spaces from the left side of the screen. Now enter the close window command:

```
SYS49440
```

and the window will disappear. Let's move the window lower on the screen by entering:

```
R%=12: SYS49432
```

The window will now be near the bottom of the screen. Try changing one or more of the window parameters and issuing a window command to see what effect this has on the window. If you enter a command and nothing happens, check that you have not tried to use a window size that is too large or located the window where it would run off the screen.

Writing directly to the window will require a few lines of BASIC. The following will do the job.

```
1000 SYS49436: REM WRITE WINDOW COMMAND
```

```
1010 GETX$: IFX$=""THEN1010
1020 IFX$=CHR$(13)THENRETURN: REM CHECK
FOR RETURN KEY
1030 IFX$=CHR$(20)THENW$=LEFT$(W$,LEN(W$
)-1: GOTO1000:REM CHECK FOR INST/DEL KEY
1040 IFX$=CHR$(147)THENW$="": GOTO1000:
REM CHECK FOR SHIFT CLR/HOME KEY
1050 W$=W$+X$: GOTO1000
```

This is written as a subroutine and can either be called from within a program or by direct command. To try this in direct mode, open a window anywhere on the screen and enter:

```
GOSUB1000
```

Any key you press should now show up in the window. The keys to clear the screen and delete characters work within the window just as they normally do on the whole screen. The RETURN key will terminate the subroutine and return control to the calling program. There is no cursor and the rules for writing to the window are the same as described above for writing window messages, i.e., SHIFTed letters will be in reverse mode, press £ key to jump to next line, and special function keys are taboo.

After you press RETURN to terminate the subroutine, the contents of the window are in the form of W$ and can be saved for later use by setting some other variables equal to W$.

MISCELLANEOUS

The five variables used in the window program cannot also be used in your program. If there is a conflict, you have the option of changing either the variables in your program or in the window program. Before you conclude there is a problem, remember that the same letter can be used three times, once for each variable type. Thus W$, W%, and W represent three separate and distinct variables to the Commodore system.

The window program is designed to make changing the variable identifiers relatively easy. The first number in the data statements is 87, which is the ASCII code for W in W$. The next four numbers consist of 128 added to the ASCII code for the letter identifiers in W%, H%, R%, and C% respectively. You can use any letter you want, but you cannot change the variable type and are restricted to single letter variables.

Both programs are relocatable and can be loaded anywhere in memory. For the 64, the variable ML in line 1 sets the point in memory where the program will start loading. If you want the program to load someplace else just set ML to the desired value before you run the program.

Changing the location of the program in the VIC is a little more involved. Delete line 1, 2, and 3 and substitute the following:

```
1 ML=XXXX: HB=INT(ML/256): LB=(ML/256-HB
)*256
```

where XXXX represents the memory location where you want the program loaded. Only the 418 byte program will be moved. The areas the program uses for data storage will remain where they were.

You can achieve some unusual effects by putting the window parameters in a loop. One example of this is the multiple windows that appear when you run the window program. You can use the program to print overlapping windows like those often used in software advertisements, but you can only "close" and restore the screen for the last window opened. ☐

Important: in order to fit the DATA statements for the 64 version on two lines each, you'll have to omit the space between the line number and the word DATA, and use the abbreviation for DATA: D [shift A].

SEE PROGRAM LISTINGS ON PAGE 97

FIDGITS

A Learning Game for the C-64

By Walter E. Meyers

If you have a preschooler learning the alphabet, or a child or seven or eight who loves computer games but has far more trouble with the keyboard than with a joystick, introduce him or her to *Fidgits*, creatures who both teach the alphabet (if needed) and show the player where the keys are on the keyboard. In the game, each of twenty-six fidgits wears a sweater with a letter of the alphabet. A randomly chosen fidgit opens its mouth and sings a note that continues to play until the corresponding key is pressed. Then another fidgit sings, and so on. After ten have sung, a new screen rewards the player with a "TEN RIGHT!" message, and asks if he or she wants to play again.

The program *Fidgits* uses a custom font with redesigned characters, some for graphics, and some for another purpose: no matter how sharp the television screen, some color combinations cause smearing of the standard font. Most of the *Fidgits* characters are used to form letters large and wide enough to allow easy readability with a variety of colors. The game also includes a number of useful graphics techniques adaptable to your own programs.

Near the beginning of the program, line 10 does something that may seem pointless: it reads the first 129 data items and does nothing with them. The reason for the strange command

```
FOR I=1 TO 129:READ A$:NEXT
```

is this: these first data items contain the frequencies of the musical notes for the "ABC" song heard while the playfield is being drawn. The song may be repeated any number of times, and therefore its data has to be at the beginning of the data listing so that the note values are right there at the head of the line after a RESTORE command resets the DATA pointer.

Then the program goes to the listing of strings beginning at line 1000. When it returns to line 100, it sets up a title page with sound and animation. One subroutine here is worth comment: lines 110 through 130 set variables for screen line (LN) and column (COL), then GOSUB to line 50:

```
POKE 781,LN: POKE 782,COL: POKE 783,0:
SYS 65520
```

Line 50 calls up a machine language routine in the kernal jump table that sets the cursor to the screen position specified. On return from the GOSUB, the program meets a PRINT statement, and the string is printed where desired, without a lot of cursor ups or downs. (For a detailed description of the subroutine, see Sheldon Leemon's *Mapping the Commodore 64*, pp. 70-71).

The next two blocks of lines, 300 to 470, first print the instructions for the game; while the instructions are on the screen, many things are happening. First, line 400 POKEs a machine language subroutine into memory starting at 49152. This subroutine, when called, copies the character set from ROM at 53248 into RAM. The ML routine locates character memory at the 2K block beginning at 14336, but the routine can be modified to put character memory wherever you want. Here's how to do it: '14336' is the number $3800 in hexadecimal notation. The ML routine POKEs this number into two zero-page locations, the left half–'38'–into 254, and the right half–'00'–into 253. These two locations together point to the place where the character set will be copied. Hexadecimal '38' equals decimal 56, and the number '56' is found in line 3010 in the data statements, the fourth item in the line.

To put character memory at any other 2K block, convert the location to hex: for example, '12288' is $3000.

Then take the left half of this number ('30' in this case) and change it back to decimal. $30 equals decimal 48. Substitute '48' for '56' in line 3010, and the character set will be copied beginning at 12288.

While the player has been reading the instructions, the character set has been copied into RAM. Lines 430 to 440 modify the character set by reading the data table beginning at line 3100, and POKEing the necessary numbers into the new character memory block.

For example: in its turn, line 3185 is reached by the program. First, the program reads "A"–185. The C-64 offers 256 characters in its character set, and a sequence of eight numbers specifies the form of the letter or number or graphics symbol. Since the character set has been copied into RAM, the program can modify the forms of the symbols. Line 3185 gives the information to modify the one hundred and eighty-fifth character. To find that character, the program follows the instructions of line 440: it multiplies 185 by 8 (the number of bytes in each character) and adds that total—1480—to the starting location, 14336. Then it POKEs the next eight numbers into the eight locations starting with 15816. This modifying continues, and when line 450 prints the prompt, it's all done.

The appearance of the instructions hasn't changed because the 64 is still using the character set in ROM: the custom font will not appear until the program tells the 64 where to look for the new character information. Location 53272 is the memory control register of the 64's graphics chip, and the '14' in line 470—

```
POKE 53272,(PEEK(53272) AND 240) OR 14
```

sets bits 1-3 in that location to point to the 2K block beginning at 14336. If you copy the character set to a different block, to the one beginning at 12288 for example, then the instruction to point to the correct block would read

```
POKE 53272,(PEEK(53272) AND 240) OR 12
```

One final point about the program: line 598, just before the main loop, disables the STOP key and the STOP/RESTORE combination. Because the game will be played by children who don't know the alphabet, we want to make sure that a hesitant finger doesn't stop the program. For the same reason, lines 630-650 set up a key trap: the GET in line 630 waits for a key to be pressed. Line 640 converts whichever key is pressed to its ASCII number. Since we have disabled the STOP and STOP/RESTORE keys, we need to check here to see if the player wants to quit. Function key #1 has been reassigned as the STOP key: its ASCII number is 133, and if function 1 has been pressed, line 640 sends the program to the ending routine. Line 650 next ignores the key that is not a letter of the alphabet. The child may hit STOP, RETURN, SPACE, a cursor key, a number, CLEAR, or anything but a letter without hampering the program. ☐

SEE PROGRAM LISTING ON PAGE 107

GATORS N SNAKES

For the C-64

BY JAMES SANDERS

Gators N Snakes is an arcade style game that utilizes a number of the great features of the C-64. What gives the program its great movement is a machine language routine that moves eight sprites at random speeds. The same routine also detects sprite to background collisions which are then handled in the BASIC portion of the program.

Gators N Snakes has as its setting the murky waters of a Louisiana swamp. The hunting party is on the last leg of the long journey. The gator and snake infested swamp is the final obstacle that separates them from the security of home. Your job is to help the hunters reach their homes by carefully crossing that dreaded swamp.

After the game is loaded and run, the playscreen becomes visible with gators and snakes swimming up and down the swamp waters. There are four returning hunters on the left side of the screen, and their destination on the right side. Sounds simple enough, right? The problem (and the object of the game) is to get the hunters home in one piece.

You have four opportunities to get them home. The gators and snakes must be avoided at all costs. Contact with them means certain death. If you are successful in getting a man to his home, you are rewarded with an additional man. If you are able to get all four men home, you advance to the next level and may try again. There are ten levels of difficulty; the increase is accomplished in two ways. The more noticeable is the increased size of the reptiles. The less noticeable but more dangerous is the gradual increase in the speed at which the reptiles swim. This speed creeps up on you, so be careful.

The program utilizes another ML routine that quickly moves the character set in order to generate a few custom characters.

When all four men have met their doom, the game is over. At this point the current score is compared to the existing high score and replaces it if it's higher. The high score is maintained for the next game (which is begun by pressing the fire button).

I have inserted two checks to insure that the ML routines are typed in correctly. Both will signal the lines in which the mistake is made in order that the mistake may be easily found and corrected.

Carefully type in the program, and save it to tape or disk before you run it. After it is saved plug a joystick into port #1 and get ready to cross the *Gator N Snake* infested swamp. □

SEE PROGRAM LISTING ON PAGE 111

PROGRAMMING TIPS WANTED

Here's a chance for you, the readers of *Ahoy!*, and us, the editors of *Ahoy!*, to work together. If we each do our part, we can make *Ahoy!*'s soon-to-debut programming tips column the finest in any magazine. Your part will be to provide hints, tips, and programming routines that can simplify your fellow Commodore computerists' lives, enhance their abilities, or both. Our part will be to come up with a catchier title by the time this column actually appears.

Keep run-of-the-mill, twice-told tips to yourselves. This is *Ahoy!* We want valuable programming tips. And we'll pay premium rates for them.

Send your submissions to *Ahoy! Programming Tips*, Ion International Inc., 45 West 34th Street—Suite 407, New York, NY 10001.

Sale

Commodore 64 Accessories

☆ ☆ SUPER ☆ ☆

Summer Sizzler Sale

SAVE

Sale Positively Ends 9-10-85

Prices will go back to regular sale prices

	List	Reg. Sale Price	Summer Sizzler Sale
C128 Commodore Computer Expandable to 512K, runs C-64, CPM, and 7.0 Programs. (Add $10 Shipping)	$349.00	$299.09	$289.95*
15½" Commodore 150-170 CPS Printer Near Letter Quality, Multiple Pin Tractor / Friction Feed. Best Printer Value in U.S.A. (Add $17.50 Shipping)	$895.00	$299.00	$249.95*
Commodore-64 IEEE Interface Allows you to run Pet Peripherals on the C-64, including the One Megabyte Disk Drive and 15½" Printer	$109.95	$69.00	$65.95
Juki Printer/Typewriter Letter Quality, daisy wheel, use as typewriter and/or printer (auto correction) (Add $10 Shipping)	$349.00	$249.00	$229.95*
SCM 80 CPS Printer Tractor/Friction 10" Famous Name Printer does Graphics w/Interface. (Add $10 Shipping)	$299.00	$159.00	$149.95*
Cardco G Plus Interface Converts Commodore to centronics for use with most printers, plus does Commodore graphics on graphic printers	$109.00	$59.00	$49.95
Alphacom 40 Column Printer Thermal technology - does graphics. (Add $10 Shipping) *Alphacom C-64 or Atari Interface $8.95.*	$99.00	$24.95	$22.95
190K Slim-Line Disk Drive Cooler, 20% faster, quieter than 1541 drive (Add $10 Shipping)	$249.00	$169.00	$139.95*
One Megabyte Disk Drive (1000K) Double sided drive hooks up to C-64 with IEEE interface, perfect as a second drive. (Add $10 Shipping)	$889.00	$199.00	$179.95*
13" Premium Quality Color Monitor Better than 1702, with separated video inputs. (Add $14.50 Shipping)	$399.00	$199.00	$169.95*
Voice Synthesizer Allows you to talk through your computer. Optional software lets you play talking adventure games (Zork, etc.)	$89.00	$49.00	$39.95
80 Column Board - 4 Slot Expander Allows you to program in 80 columns, plus gives you a 4 slot switch selectable expander	$149.00	$79.00	$59.95
Auto Dial Modem with Super Smart Software Upload, Download, Print, Catalog, This Package has it all!!! Best Modem Value in the U.S.A.!!	$129.95	$59.00	$39.95
Musicalc I Software The Best Musical Software for the C-64, allows you to change all parameters.	$59.95	$29.95	$14.95
Oil Barrons Software Better than Monopoly, comes with game board, disks and instruction manual. Strike Oil or Live in the Poor Farm	$49.95	$19.95	$9.95

*** plus Software Coupon Discounts**

Add $3.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $6.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

SALE 80 CPS PRINTER SALE

Smith Corona

• Commodore 64 • Atari • Apple • More

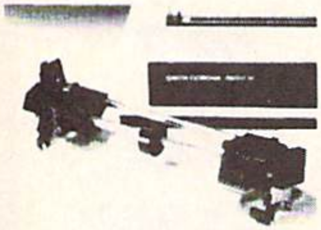
• 11" Carriage • Friction/Tractor • Graphics

Super Summer Sale
Ends 9-10-85

$149

1 Year Immediate Replacement Warranty





Tractor Feed Included FREE!

SMITH CORONA

The Fastext 80 dot matrix printer from Smith-Corona. A high speed, high flexibility printer with a low suggested manufacturer's retail price of only $299.00. **Sale $149.00.** It does your graphics with ease and prints letters with speeds up to **80 CPS**. ***(Does Commodore graphics with graphics interface.)*** As for flexibility, it has bi-directional printing, friction and tracter feed . Plus six different pitches including condensed and enlarged type for impeccable characters in a variety of sizes. Also standard is a full line buffer and thrifty drop-in ribbon cassettes that yield up to a million characters. Finally, it's compatible with virtually all personal computers. Teamed up with your personal computer, it'll keep track of your transactions, fly through finances, help with homework, even plan the family menu. All this with the assurance of superb Smith-Corona quality built in, makes the Fastext-80 a lot of dot for your dollar. List $299.00. **Super Summer Sale $149.00.**

SPECIFICATIONS:

Size Weight: Height 4" Width 16.5" Depth 9" Weight 8.2 lbs.
Electrical Needs: 120V 60Hz
Internal Char. Coding: ASCII/ISO
Print Buffer Size: 132 Bytes (1 line)
No. of Char. in Char. Set: 96 ASCII/ International
Graphics Capability: Standard 60, 72 Dots Per Inch Horizontal, 72 Dots Per Inch Vertical
Pitch: 10, 12, 16.7, 5, 6, 8.3
Printing Method: Impact Dot Matrix
Char. Matrix Size: (Standard) 9H x 8V, (Elongate) 10H x 8V
Printing Features: Bi-directional, Short line seeking
Printing Speed: 80 CPS

PAPER

Type: Plain
Forms Type: Fanfold, Cut Sheet
Max Paper Width: 11"
Feeding Method: Friction Feed Std.: Tractor Feed Included

RIBBON

Type: Cassette — Fabric inked ribbon
Life: 1 million characters

CHARACTER MODE

Character Font: 9 x 8 Standard, 10 x 8 Elongated, No. 8 pin to be used for underline
Character Set: 96 ASCII, 11 x 7 International Char.
Pin Graph Mode: The incoming bit pattern corresponds to the 8 pins in the print head
Resolution: Horizontal: 60, 72 dots/inch, Vertical: 72 dots/inch

PARALLEL INTERFACES

Commodore 64 — $39.00 Apple II, IIe, II+ — $59.00 Atari — $59.00

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES, EXCEPT CANADA.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA — MASTER CARD — C.O.D.
No C.O.D. to Canada, APO-FPO.

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 179

COLOR MONITOR

Sale

SALE!!!

16 COLORS



13" Color Computer Monitor

(Premium Quality)

- **Built in Speaker and Audio**
- **Front Panel Controls**
- **For Video Recorders**
- **For Small Business Computers**
- **Apple - Commodore - Atari - Franklin - etc.**

Add $14.50 shipping, handling and insurance. Illinois residents please add 6% tax. We do not ship to foreign countries (including Canada), Alaska, Puerto Rico, APO-FPO.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!

(Premium Quality)

- **Beautiful Color Contrast**
- **High Resolution**
- **Separated Video**
- **Sharp Clear Text**
- **Anti-Glare Screen**
- **40 Columns x 24 Lines**
- **Supports 80 Columns**

List $399.00

SALE $169.00*

*Connecting cable $9.95

15 Day Free Trial - 90 Day Immediate Replacement Warranty

14" COMMODORE 1702 COLOR MONITOR
(add $14.50 shipping)

List $299.00 — Super Summer Sizzler Sale **$179.00***

12" XTRON SUPER HI-RESOLUTION MONITOR
80 Columns x 24 Lines, Super Hi-Resolution 1000 lines Green or Amber Super-Clear "Easy to Read" text with special anti-glare screen!

List $249.00 — Super Summer Sizzler Sale **$109.95***

12" ZENITH HI-RESOLUTION GREEN OR AMBER TEXT DISPLAY MONITOR
80 Columns x 24 Lines, Hi-Resolution, crisp clear easy to read text with anti-glare screen! A MUST for word processing.

List $199.00 — Super Summer Sizzler Sale **$89.95***

12" MONITOR GREEN OR AMBER TEXT
80 Columns x 24 Lines, easy to read up front controls

List $159.00 — Super Summer Sizzler Sale **$69.95***

* Connecting cable $9.95

• LOWEST PRICES • 15 DAY FREE TRIAL • 90 DAY FREE REPLACEMENT WARRANTY
• BEST SERVICE IN U.S.A. • ONE DAY EXPRESS MAIL • OVER 500 PROGRAMS • FREE CATALOGS

Add $10.00 for shipping, handling and insurance. Illinois residents please add 6% tax. Add $20.00 for CANADA, PUERTO RICO, HAWAII, ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars. WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.
Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery, 2 to 7 days for phone orders, 1 day express mail!
VISA MASTER CARD — C.O.D. No C.O.D. to Canada, APO-FPO

PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

Reader Service No. 180

SCREEN DUMP ROUTINE
FOR COMMODORE 64 AND PET COMPUTERS
By Alfred J. Bruey

It is often true that the hardest part of writing a program is writing the documentation. It is also often true that the most tedious part of writing documentation, especially if a program is menu-driven, is typing up the menu screens so that they resemble the screens that the user will see as he runs the program. This is also the most expensive part of the typesetting function if you are preparing your documentation for publication.

INCORPORATING THIS PROGRAM INTO YOUR SYSTEM

To use the program shown on page 100, just type in the program you wish to document, making sure to leave open the line numbers from 50000 to 50250 (although you can renumber this routine any way you want). After you've entered your program, append the routine on page 100 to the end of it.

AN EXAMPLE

You can use this program to demonstrate itself. Just enter the screen print routine and add the BASIC line

```
10 GOSUB 50000: STOP
```



"Computer, computer, look at my face, Am I the fairest in your database?"

The program shown on page 100 is for the Commodore 64. To make it run on the PET/CBM models, change line 50040 to

```
50040 FOR I = 32767 TO 33750 STEP 40
```

Then type LIST so that the program will be displayed on the screen. Now type RUN and press return. Whatever is on the screen will be reproduced on your printer.

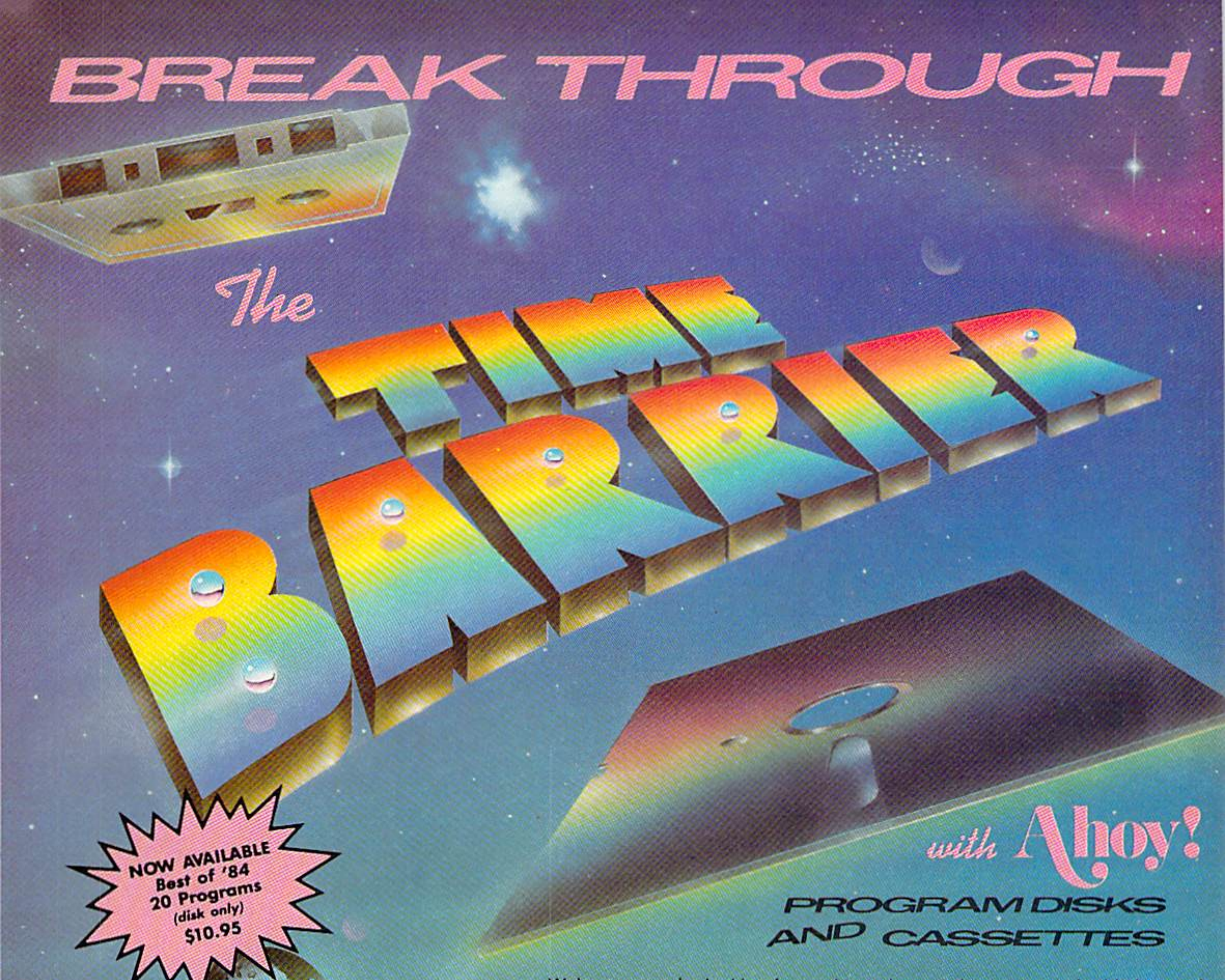
HOW TO USE THIS PROGRAM

At any time during the running of a program when you want to keep a hardcopy (printed) record of what appears on the screen, just put in a GOSUB statement to execute this subroutine. The program will pause long enough print out the screen image and then continue with its processing. You will find that the screen images are faithfully reproduced with one exception: all lowercase alphabetic characters will be converted to uppercase. This program will print all upper case and graphics characters, both regular and reversed. Once you've printed copies of all the screens for your documentation, you can turn the GOSUBs into REM statements or else remove them and the subroutine from the program completely. A sample program layout might be as follows:

```
program initialization
.
.
.
coding to display first menu
GOSUB 50000
INPUT or GET statement for first menu
processing
.
.
.
coding to display second menu
GOSUB 50000
INPUT or GET statement for second menu
processing
.
.
.
```

A FINAL NOTE

You can also leave this routine in a production program to make a record of your transactions. For example, assume you are using your computer to enter requests based on customer phone calls. You can install this routine in your program in such a way that each time a call is taken and the data recorded on the screen, a hardcopy of the transaction will be printed as part of your daily records. □ ***SEE PROGRAM LISTING ON PAGE 100***



We're sure you're looking forward to trying out the programs in this issue of *Ahoy!* But we're equally sure that you're *not* looking forward to typing them in. If you're an average typist, that should take you upwards of 25 hours. Not counting time spent correcting your typing errors, of course. How long that will take is anyone's guess. • How would you like someone to type the programs for you? At a price of, say—32¢ an hour? Wouldn't it be worth 32¢ an hour to free yourself up for more pleasant pursuits—like enjoying the rest of your *Ahoy!* magazine? • If you order the *Ahoy!* Program Disk or Cassette, you'll be getting that service for just under 32¢ an hour. Because for $7.95 (postage included)* we'll mail you all the programs in this issue, on a disk or cassette that's tested and ready to run with your C-64 or VIC 20. If you subscribe, you'll save even more: 12 monthly disks or cassettes for $79.95, 24 for $149.95 (postage included)*. Back issues are available as well. • You already know how to type. Why not use the hours you spend with your Commodore to learn something new?

*Canadian and foreign prices higher.

Ahoy! uses only **maxell** *premium quality disks.*

SINGLE ISSUE AND SUBSCRIPTION PRICES

(Postage and Handling Included)	U.S.A.	CANADA	ELSEWHERE
❑ August '85 *Ahoy!* Disk ❑ August '85 *Ahoy!* Cassette	$ 7.95	$ 10.00	$ 12.00
❑ 12-Month Disk Subscription ❑ 12-Month Cassette Subscription	$ 79.95	$ 99.95	$124.95
❑ 24-Month Disk Subscription ❑ 24-Month Cassette Subscription	$149.95	$179.95	–
❑ *Best of '84* (disk only)	$ 10.95	$ 10.95	$ 10.95

BACK ISSUES $8.95

(except months listed)

In Canada add $2.00 per issue; outside US and Canada add $4.00 per issue.

❑ Jan. '84	$9.95	❑ ______	$8.95
❑ ______	$8.95	❑ ______	$8.95
❑ ______	$8.95	❑ ______	$8.95
❑ ______	$8.95	❑ ______	$8.95

SPECIFY ❑ *DISK* ❑ *CASSETTE*

Make check or money order payable to **Double L Software.** N.Y. State Residents must add 8¼% sales tax.

Send coupon or facsimile to:
***Ahoy!* Program Disk/Cassette**
Ion International Inc.
45 West 34th Street, Suite 407
New York, NY 10001

NAME ______________________

ADDRESS ______________________

CITY ____________ **STATE** ______ **ZIP** ______


MASTERTR
ARCADE-STRATEGY-ACTION
Sure Chart Winners
VEGAS JACKPOT
One of the most faithful computer simulations ever produced of a fruit machine. It incorporates the hold, nudge, gamble and shuffle features found in the real thing.
SPACE WALK
A satellite retrieval game that includes 16 levels of "low pressure" death awaiting the unskilled astronaut.
BMX RACERS
A sure chart winner. You will need skill and fast reflexes to beat this one! You can ride on 5 different courses, each course ranges from 8 to 25 different scrolling screens.
CHILLER
Opened to Five Star Reviews in Europe recently this is surely one of the most challenging and unusual games available for the C64. Written by "Commodore Programmers of the Year" Richard and David Darling. The game has five different spectacular screens with hidden twists — If you get this far you have to retrace your steps through another five of increasing difficulty.
MIND CONTROL
Another unusual arcade — strategy game in which the players strive by weird and wonderful techniques to achieve a state of miniaturisation and hence enter Zyco's brain and the game continues in another dimension.
MAGIC CARPET
An imaginative three screen multi level arcade adventure that uses very high resolution graphics and will test the abilities of the most experienced games players.
1985 THE DAY AFTER
A scrolling multi screen game that takes Commodore graphics to a 'new' state of the art level previously seen only in a handful of other games. Playability is a keyword on this 'five star game'!
KIKSTART
A one or two player game with split screen scrolling graphics that realistically simulates off road riding on eight different and challenging courses.
More Adventures of BIG MAC the Mad Maintenance Man
Multi screen arcade action with real twists needing a steady hand and a bold strategic game play. It will keep your interest alive for hours.
challenger
Neat variation on a city bomber theme using excellent scrolling graphic routines that is extremely addictive.
MAGRA and GAMES CREATOR $19.99
$9.99
SPECTACULAR
HIGH
QUALITY SOFTWARE

ONIC!

BRINGS YOU BRAND NEW
UNBELIEVABLE
COMMODORE 64 DISKS



RedHot PRICES!

$9.99

Look for MASTERTRONIC programs at your local computer dealer or mass merchant. For more information write or call (301) 695-8877

Mastercard and Visa accepted.

MASTERTRONIC INTERNATIONAL INC.
7311B Grove Road, Frederick, Maryland 21701

Reader Service No. 232

Art Gallery



Top left: *Pond* by Dave Moroz-Henry (Downington, PA), drawn with *Koala*. Mr. Moroz-Henry claims to be a genuine starving computer artist seeking gainful employment in the computer graphics or software development field. Correspondence from interested parties will be forwarded to Mr. Moroz-Henry.

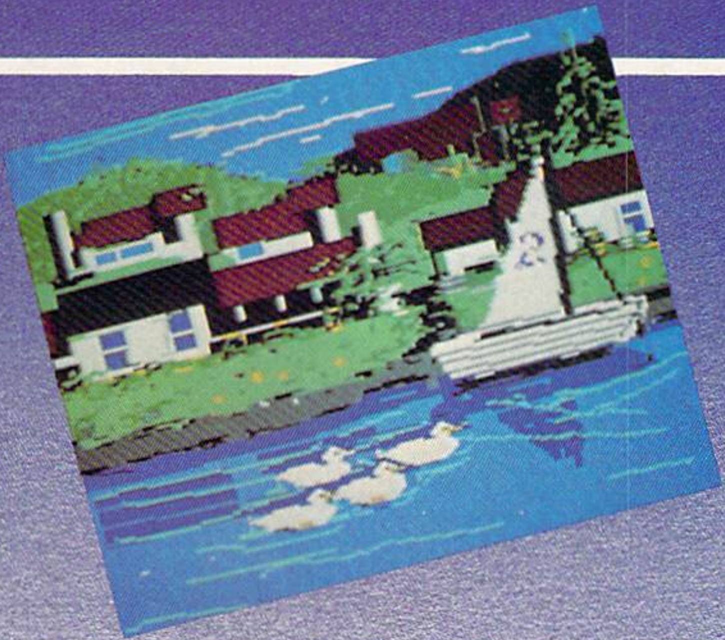






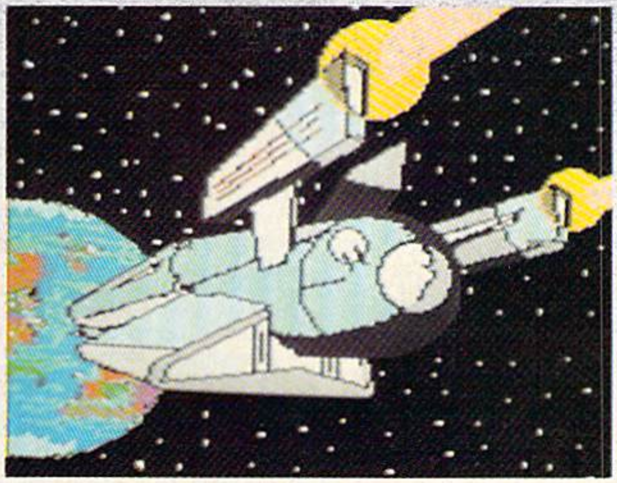
Four images by Armand Suarez (New Iberia, LA), whose work can be described as Far East or Far Out: *Fuji-San*, *Geisha* (left), *Torii* (above), and *Starship* (right). All images created with the KoalaPad and *Koalapaint* software.

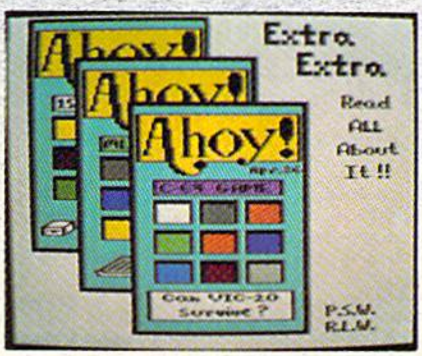
Contributors to *Ahoy!*'s Art Gallery will receive royalties based on the sale of disks containing the best computer graphics received, both published and unpublished. Send your best work on disk, accompanied by a stamped and self-addressed mailer, to Morton Kevelson, P.O. Box 260, Homecrest Station, Brooklyn, NY 11229. Indicate the drawing package that was used to create the image. If you employed a bit map of your own design, indicate the appropriate file parameters, i.e., hi-res or multicolor, location of bit map, screen or color data.



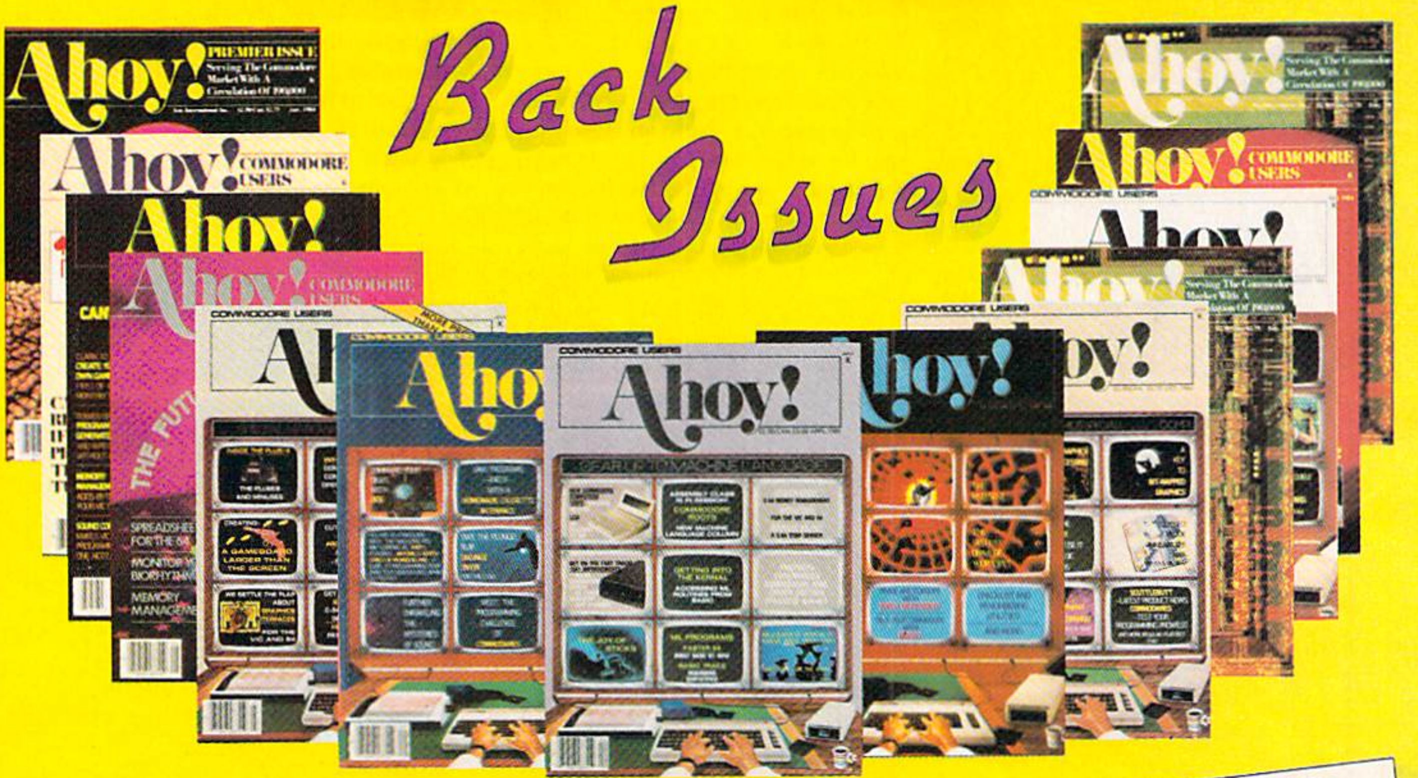


Top left: *Sailboat* by Earl Hamner (Milpitas, CA). Middle: *BC Gasson* by Peter T. Hughes (Canton, MA) is Gasson Hall at Boston College drawn in meticulous detail with *Paintpic* without zoom feature. Bottom right: an irresistible entry from Richard and Pamela Winters (Shreveport, LA).





Back Issues



ISSUE #1–JAN. '84 $4.00
64 v. the Peanut! Telecommunications! Tales of the Commodore! And ready to enter: Multi Draw 64 Graphics System! Interrupt Music Maker/Editor! A Peek at Memory! Sequential Files!

Sorry...Issues #1, #2, and #7 sold out! Reprint editions (programs and feature articles only) available for $4.00 each.

ISSUE #2–FEB. '84 $4.00
Illustrated tour of the 1541! Artificial intelligence! Synapse's Ihor Wolosenko interviewed! And ready to enter: Music Maker Part II! Screen Manipulation! Night Attack! Relative Files!

ISSUE #3–MAR. '84 $4.00
Anatomy of the 64! Printer interfacing for VIC & 64! Educational software series begins! And ready to enter: Address Book! Space Lanes! Random Files on the 64! Dynamic Power!

ISSUE #4–APR. '84 $4.00
Petspeed and Easy Script tutorials! Printer interfacing continues! Laserdisc! And ready to enter: Apple Pie! Lunar Lander! Name that Star! Lower case descenders on the 1525 printer!

ISSUE #5–MAY '84 $4.00
Future of Commodore! Inside BASIC storage! Memory management on the VIC & 64! Guide to spreadsheets! And ready to enter: Math Master! Air Assault! Biorhythms! VIC Calculator!

ISSUE #6–JUNE '84 $4.00
Game programming column begins! Program generators! Rupert on inputting! Memory management continues! And ready to enter: Post Time for the 64 & VIC! Alpiner! Sound Concept!

ISSUE #7–JULY '84 $4.00
MSD dual disk drive! Database buyer's guide! Training your cursor! Screen displays! File Sleuth! Users Groups! And ready to enter: Renumbering! Checklist! Math Defender! Brisk!

ISSUE #8–AUG. '84 $4.00
Choosing a word processor! Computational wizardry! Creating your own word games! Sound on the 64! And ready to enter: Micro-Minder! Directory Assistance! The Terrible Twins!

ISSUE #9–SEPT. '84 $4.00
Program your own text adventure! Build a C-64 cassette interface! Video RAM! And ready to enter: Salvage Diver! DOS! Sound Explorer! The Castle of Darkness! Base Conversions!

ISSUE #10–OCT. '84 $4.00
C-64 graphics programs! Bit-mapped graphics! Joystick programming! And ready to enter: VIC 40 Column Operating System! BAM Read & Print! Emerald Elephant! Lawn Job!

ISSUE #11–NOV. '84 $4.00
Music programs & keyboards for the 64! Graphics feature continues! And ready to enter: PTE word processor! Block Editor! Alternate Character Set for the 64! The Tunnel of Tomachon!

ISSUE #12–DEC. '84 $4.00
Buyer's guide to printers! 1525 printer tutorial! Custom characters! User Guide to KMMM Pascal! Diving into BASIC! And ready to enter: Construction Co.! Space Patrol! Cross Ref!

ISSUE #13–JAN. '85 $4.00
VIC/64 OS exposed! Sprites! 1541 device # disconnect switch! Ghostbusters! And ready to enter: Ultra Mail! Music Tutor! Alice in Adventureland! Midprint! To the Top! Tape/Disk Transfer!

ISSUE #14–FEB. '85 $4.00
Printer interfacing! Multicolor sprites! Modems! Bulletin boards! Theory of game design! And ready to enter: Futurewar! Fontasia! VIC Eraser! Insurance Agent! Flankspeed! Telelink 64!

ISSUE #15–MAR. '85 $4.00
Making multiscreen gameboards! Inside the Plus/4! Commodore DOS! And ready to enter: Old Routine! Programmable Functions! Automatic Line Nos.! Home Budget! Salmon Run!

ISSUE #16–APR. '85 $4.00
Assembly language column begins! Programming the joystick! 1541 disk drive alternatives! And ready to enter: Hop Around! Faster 64! Booter! Elecheck! BASIC Trace! Space Hunt!

ISSUE #17–MAY '85 $4.00
Disk drive enhancements! Install a reset switch! Assembler escapades! And ready to enter: Super Duper! Two-Column Directory! DSKDU! Raid! DOS Plus! Font Editor! Tile Time!

ISSUE #18–JUNE '85 $4.00
Music & graphics entry systems! How modems work! Inside the 6510! And ready to enter: Quad-print! Mapping 4.4! Towers of Hanoi! Speedy! Duck Shoot! The 6510 Simulator!

ISSUE #19–JULY '85 $4.00
PROM programming! 3-part harmonies on VIC/64! Speeding pixels! And ready to enter: Auto-Append! Script Analysis! Wizard of Im! Lucky Lottery! Brainframe! Etch! Printat!

STOP PROGRAMMING...

...until you have a complete collection of *Ahoy!* behind you! Our articles and tutorials can save you hours of toil at your Commodore keyboard...our programs, hundreds of dollars at the software counter! Use the form below to order your issues while limited supplies last!

Use coupon or facsimile. If ordering more than three issues, list choices on separate sheet.

If you have a modem and want more complete information on any of the back issues listed, call *Ahoy!*'s Bulletin Board Service at 718-383-8909.

Ahoy! BACK ISSUES
ION INTERNATIONAL INC.
45 West 34th Street
Suite 407
New York, NY 10001

Please Send Me The Following:
_____ Copies of issue number _____
_____ Copies of issue number _____
_____ Copies of issue number _____

Enclosed Please Find My Check or Money Order for $__________
(Outside the USA please add $1.00 for every copy)

NAME ____________________
ADDRESS ____________________
CITY __________ STATE __________ ZIP CODE __________



Javier Romero

Falling Needles and Shared Birthdays

Simulations of the Real World By Dale Rupert

Last night my Commodore 64 dropped nearly a quarter of a million needles on a hardwood floor. It kept track of how many of those needles fell across a crack, and thereby proclaimed that the value of pi is 3.15845536. My Commodore 64 also calculated that with at least 117 people in a room, there is a one hundred percent probability that two or more of them have the same birthday.

Unfortunately, neither one of these conclusions is totally correct. Both conclusions are somewhat crude approximations to the truth. This month we will look at some methods and limitations of simulating real-world events on a computer.

Three basic ways of predicting outcomes of events are calculation, enumeration, and random simulation. Calculations are readily performed if a formula and the relevant values are known. Probable outcomes of statistical experiments involving dice, cards, and other such devices of chance are often determined by enumeration. Enumeration is a process whereby all possible outcomes are

listed, and the successful outcomes are simply selected from the list and counted. Within limits, calculation and enumeration on the computer can arrive at an exact result, if one exists.

A standard method of computer simulation is to randomly select the outcome of a series of events such as the roll of the dice or the location and orientation of a needle dropped on the floor. By analyzing the randomly chosen results, the likelihood of a particular outcome may be determined. The results of the simulation will be only as good as the randomness of the selection and the accuracy of any calculations involved.

Before we begin simulating various events, we will define some terms and concepts. It's time for a short course in probability and statistics.

CHANCES ARE...

Let's begin with a simple example that everyone can readily visualize. To determine the probability of getting two heads when two coins are flipped, proceed as follows. If we flip two coins, there are four possibilities: HH, HT, TH, and TT, where H stands for heads and T stands for tails. We have thus enumerated all four possible outcomes for the flipping of two coins. Now we merely count the number of successful outcomes, namely the number of outcomes containing two heads.

Obviously only one out of the four possible outcomes is successful. Consequently we conclude that the probability of getting two heads when flipping two coins is one out of four or one fourth or twenty-five percent. The probability of getting at least one tail can be found the same way: count the number of successful outcomes and divide by the total number of possible outcomes. From the list above, there are three outcomes which contain at least one tail. Therefore the probability of getting at least one tail is three out of four or seventy-five percent.

The implications of statistics and probabilities are often misunderstood. Indeed the probability of getting two heads when tossing two coins is twenty-five percent. That means in the long run two heads will be flipped one fourth of the time. A twenty-five percent probability does not mean that after three unsuccessful flips, the fourth will be successful. In fact the first fifty flips (or the first million flips for that matter) could conceivably be all heads, but such a sequence is not likely to happen often.

If more than a half dozen sets of double heads occurred in a row, there would be reason to believe that maybe something is fishy (loaded coins or loaded flipper perhaps). It is a straightforward process to calculate the probability that six sets of double heads would occur in a row. Since each outcome is independent of every other one, simply multiply the individual probabilities together. The result is one fourth raised to the sixth power, or 0.000244. Probabilities are frequently given as percentages, so multiply the result by 100 to get 0.024 percent. That's why we might be suspicious if six sequential pairs of heads were flipped more than rarely.

The probability of success and the probability of failure add up to one. If the probability of getting two heads is one fourth, the probability of not getting two heads is three fourths. It is easy to see this if you count the number of "failures" in the enumerated list and see that there are three. The percent probabilities of success and failure add up to one hundred, so the probability of *not* getting six pairs of heads in a row is (100 − 0.024) or 99.976 percent.

Probabilities deal with successful outcomes as a fraction of total possible outcomes. Odds are related but slightly different. Odds compare the number of successful outcomes with the number of failures. The odds that a pair of heads will be flipped are one to three (one chance in favor versus three chances opposed). We will deal with probabilities rather than with odds.

Now that we have the rules and the vocabulary for dealing with probabilities, let's drop some needles on the floor.

FALLING NEEDLES

Count Georges-Louis Leclerc Buffon was an eighteenth-century French naturalist and experimentalist. His experiment which we will simulate involved dropping needles onto a ruled surface and counting how many of them landed on a line. Surprisingly enough, he found that the proportion of needles landing on a line was a multiple of pi! Specifically, if the lines on the surface are a distance H apart, and the length of the needle is L, the prob-

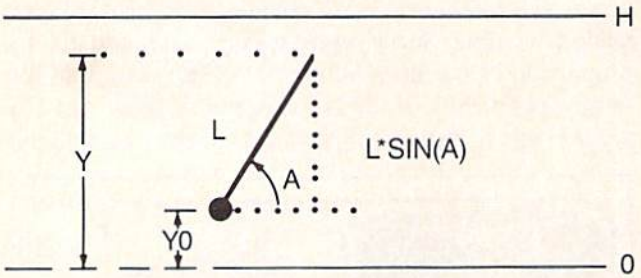
ability that a needle falls on a line is two times L divided by the quantity H times pi. An account of Count Buffon's experiment, as well as multitudes of other interesting problems, are presented in the book *Mathematics and the Imagination* by Edward Kasner and James Newman (Simon and Schuster). Try a bookstore or a library for this most entertaining volume.

How can a computer simulate the dropping of needles? A little head-scratching reveals that there are essentially two random parameters: the location of one end of the needle and the location of the other end. The two parameters are related. In fact, if we pick the location of one end and the angle of orientation of the needle, we can calculate where the other end is.

Assume that each board of a hardwood floor has a width H and a needle has a length L. One end of the needle may fall at any distance Y0 from one crack. If the needle makes an angle A with respect to the horizontal, the other end of the needle is at a distance Y from the crack, where

```
Y = Y0 + L * SIN(A)
```

The diagram below should help.



A needle of length L is dropped onto a floorboard of width H. Will it land on a crack?

Some thought should convince you that we lose no generality by using a single floorboard of width H. We will choose a random value for Y0 between 0 and H. We will also choose a random value for the angle A. Deciding what limits to place on the size of A was somewhat difficult. Let's consider a needle pointing horizontally to the right to have an angle of 0 radians (0 degrees) and a needle pointing vertically upward to have an angle of pi/2 radians (90 degrees). We should be able to limit the range of A from 0 to 2*pi radians (0 to 360 degrees) to cover every possible orientation. To do so would require that we use the value pi in our random number generation, and that is the very value we are trying to find.

Consequently we will choose a random value for A between 0 and some arbitrarily large number, say 30000. Although 30000 radians is not an exact number of complete revolutions, our results should not be overly biased. You might try 31416 radians as an upper limit which is even closer to an integral number of revolutions.

The groundwork is nearly complete. The computer will pick two random numbers, Y0 and A. It will calculate the location of the other end of the needle, Y. Then it must decide whether the needle lies across a line. From the diagram you can see that the needle does lie across a line if Y is greater than H or if Y is less than zero. The program *Needles* (see page 90 in the program listings section) implements this simulation.

The length of the needle is chosen to equal the width of the floorboard. K is the constant relating the ratio of L and H to the value of pi. TRIES keeps track of the number of needles dropped. WINS is incremented whenever the conditions in line 90 are met, that is, whenever the needle falls across a line. The variable PI is the calculated result, which should theoretically equal pi if enough needles were randomly dropped. TTL and AVG keep a running average of the values of PI after each drop. The value of AVG will change less rapidly than PI and might be easier to follow.

My Commodore 64 dropped needles for nearly eight hours. After 231,878 drops, it showed a value for PI of 3.15845536 and a value for AVG of 3.15426395. These are certainly not very impressive results. Kasner and Newman mentioned in their book that Lazzerini, an Italian mathematician, dropped 3,408 needles and came up with a value for pi of 3.1415929, correct to six decimal places! (Pi is approximately 3.141592654.)

Why are the results of this program so inaccurate? The most likely suspect is the random number generator itself. If you ran the random number generator program in the June 1985 *Rupert Report*, you saw that there are amazingly obvious patterns among the supposedly random numbers being generated. The only other factors which could account for such a poor showing are the limited word length for numeric variables and the inaccuracies of the mathematical operations. Perhaps using a different range of values for A would make a difference.

Let me know of any modifications you make to this program and what your results are. Maybe using a different random number generator would help. Your ideas are welcome. Let's move on to another simulation.

MATCHING BIRTHDAYS

Perhaps you have heard that with twenty-three people or more in a group, the odds are better than even (there is greater than 50% probability) that at least two people will have the same birthday. Once again this means that if you investigate a great enough number of groups, slightly more than half of them will contain at least one set of matching birthdays.

There is a formula for calculating the probability of matching birthdays. We will derive the formula and write a program to evaluate it for groups of any size. Next we will write a program to randomly choose birthdays for people in groups of a given size. It will be up to you to see how closely the results of the simulation program match the calculated results.

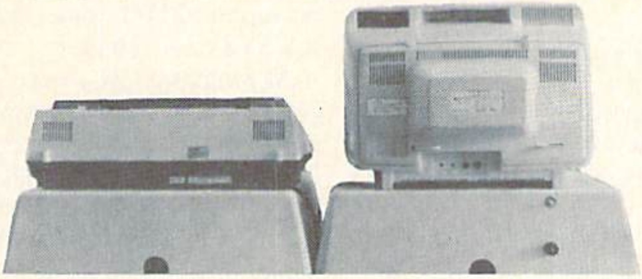
Let's consider a group of three people. To calculate the probability that at least two of them have the same birthday, we will work in reverse. We will calculate the probability that no two of them share a birthday. We will subtract that result from one, since the probability of a failure plus the probability of success add up to one.

Consider the first person. His birthday may be any one of 366 days. (We are assuming all birthdays are equally distributed throughout the population, which in actuality they aren't.) If he were the only person in the group, the

INNOVATIVE ORGANIZERS, INC.

For Commodore 64™ & VIC-20™





WELCOME, to the family of **INNOVATIVE ORGANIZERS.** These **ATTRACTIVE** power organizer stands, made of **ABS PLASTICS, U.L. RATED,** provide an opportunity for a more **PROFESSIONAL** looking computer installation, both at **HOME** and in the **WORK PLACE.**

There is **AMPLE** room under the organizer to place all the computer accessories: **POWER SUPPLY, MODEM, GAME CARTRIDGES, MEMORY EXPANSION,** and **MUCH MORE.**

Now, let's examine how these organizers work. There are 3 CRT-Models.

CRT-1 holds a Monitor or T.V. Set and a VIC 20 or 64. All wires & cartridges fit neatly under the organizer.

CRT-2 same as CRT-1, but also has: **ONE-STEP MASTER SWITCH** (permits 4 pieces of hardware to be turned on and off from one switch, plus circuit protection, incoming power fused with 6 AMP Fuse).

CRT-3 same as CRT-2, but also has: **SURGE & SPIKE PROTECTION** built in.

P-D-1: **HOLDS 1525 PRINTER & 1541 DISK DRIVE.** All wires & cartridges fit neatly under the organizer.

Ask about our Power Lock Switch & Power Portable Computer Fan & 2 Outlet Surge & Spike Protection.

I.O.
INNOVATIVE ORGANIZERS
YOUR ALTERNATIVE
INNOVATIVE ORGANIZERS

CALL FOR LIST OF SOFTWARE AVAILABLE

VIC 20 and Commodore 64 are trademarks of Commodore Electronics, L.T.D.

P.O. Box 108, Markleville, IN 46056

CRT-1	$17.98
CRT-2	$29.97
CRT-3	$37.97
P-D-1	$19.48

Optional Accessories

SS-1 Surge & Spike Suppressor $29.95

CF-1 Computer Fan $29.95

Plus Shipping

CALL TOLL FREE
1-800-233-3645 Ext. 15

Reader Service No. 222

probability of a failure would be one, since obviously he would not share a birthday with anyone else in the group. The probability of success for one person is zero. "Success" will mean "two or more people in a group sharing a birthday." "Failure" will mean "no two people in a group sharing a birthday."

For the second person, there are only 365 of the possible 366 days that lead to a failure, since his birthday must not match the first person's birthday. For the third person, there are only 364 of the possible 366 days that lead to a failure, since the third birthday must not match either of the other two.

The overall probability of failure for a group of three people is

$$\frac{366}{366} * \frac{365}{366} * \frac{364}{366}$$

where each fraction is the ratio of the number of failures to the total number of possibilities. To calculate this type of expression, alternate between the divisions and multiplications for best accuracy. The probability of failure is roughly 0.992 or 99.2 percent. That means the probability of success is (1 − 0.992) or 0.008 which is 0.8 percent.

Continuing this line of reasoning, we arrive at a general formula for the probability of failure for a group of NP people:

366 * 365 * 364 * ... * (367−NP) / (366) ↑ NP

As before, the probability for success is one minus that quantity. You can see that if there are 367 people in the group, the probability of failure equals 0, since the last term in the numerator is zero. That's exactly what is expected, since there are not 367 different birthdays.

The *Birthday Calc* program on page 90 consists of two parts. The first part asks for the number of people in the group, and it calculates the probability of success by evaluating the above formula with a FOR-NEXT loop. If you enter a non-integral value or a value of zero, the program branches to line 200. The second part of the program at line 200 evaluates the formula for each value of NP from 1 to 366 and displays the results one screenful at a time. VIC 20 users may modify lines 230 and 240 to tidy up the display.

If you want to print out the results, you may insert line

```
205 OPEN 4,4
```

Also change lines 230 and 240 to contain "PRINT#4," instead of "PRINT" and delete lines 270 and 280. Notice that the probability of shared birthdays in a group of 58 people is greater than 99 percent. It's still hard to believe, isn't it?

One final project this month is the program called *Birthday Simulator* (see page 90). The listing uses 24 for NUMP, which is the number of people in each group. You may change this value in line 30. For each person, a random number from 1 to 366 is chosen to represent his birthday. The array BDAY(PERSN) contains the birthday value for each value of PERSN. If the first person's randomly chosen birthday is 5 (January 5th), BDAY(1) equals 5. Similarly if BDAY(3) equals 366, the third person in the group was born on day 366 (December 31).

The other array CT( ) keeps track of the number of times a particular birthday has occurred within a group. For example, if CT(366) equals 2, then two people in the group have a birthday on December 31. Although there is an array element in CT for each of the 366 days, we only need to look at those elements corresponding to birthdays which have been assigned. The assigned birthdays are contained in the BDAY( ) array. Consequently the program uses BDAY(PERSN) as an index into the elements of CT( ) which are non-zero.

Try a simple example to understand this data structure. Suppose there are three people in the group. The first and third persons' birthdays are January 10 (DAY = 10) and the second person's birthday is January 5 (DAY = 5). The non-zero values of BDAY( ) and CT( ) are as follows:

```
BDAY(1) = 10; BDAY(2) = 5; BDAY(3) = 10
CT(5)=1 ; CT(10)=2
```

The program loop starting at line 140 to check for duplicate birthdays looks only at elements of CT( ) corresponding to valid BDAY values. Program execution is thereby much faster than if every value of CT had to be checked.

As soon as a CT( ) value greater than or equal to two is found (implying that at least two people share that birthday), lines 160 and 170 are executed. The SUCCESS flag is set to one in line 160. The loop parameter PERSN is set to the upper limit of the loop so when line 180 is executed, the loop is finished. The program quits searching for matching birthdays as soon as it finds the first one.

Line 190 tests the SUCCESS flag. If it is one, the total number of successful groups TTL is incremented. Lines 200 and 210 display the current percent probability of success based upon all groups simulated so far. Line 220 increments the number of groups and the program is repeated for a new group. SUCCESS is reset to 0 and the non-zero values of CT( ) are zeroed.

Eventually the results of this program should approximate the results calculated in the previous program. Once again, the non-randomness of the random number generator will not give ideal results, but the results should be close to the expected values.

This type of random number simulation is actually used in research to estimate values for quantities which are very difficult or even impossible to calculate exactly. You can read more about computer simulation in the *McGraw-Hill Encyclopedia of Science and Technology* and other encyclopedias under the topics of "Simulation" and "Monte Carlo Methods."

Next month we will compare the techniques of enumeration and random simulation as we analyze some tumbling dice. What is the probability of rolling "an eleven" with three dice? Working out such a problem by hand is quite a chore. We'll let the computer figure it out next month. ☐ *SEE PROGRAM LISTINGS ON PAGE 90*

COMMODARES

PROGRAMMING CHALLENGES

BY DALE RUPERT

Each month, we'll present several challenges designed to stimulate your synapses and toggle the bits in your cerebral random access memory. We invite you to send your solutions to:

Commodares, c/o *Ahoy!*
P.O. Box 723
Bethel, CT 06801

We will print and discuss the cleverest, simplest, shortest, most interesting and/or most unusual solutions. Be sure to identify the *name* and *number* of the problems you are solving. Also show sample runs if possible, where appropriate. Be sure to tell what makes your solutions unique or interesting, if they are. Programs on diskette are welcome, but they must be accompanied by listings. You must enclose a stamped, self-addressed envelope if you want any of your materials returned. Solutions received by the middle of the month shown on the magazine cover are most likely to be discussed, but you may send solutions and comments any time. Your original programming problems, suggestions, and ideas are equally welcome. The best ones will become *Commodares*!

Problem #20-1: Screen Scramble II

This is a variation on the problem suggested last month by Jesus Geliga Torres (Aguadilla, PR). Write a program which transforms the image on the screen into a new image such that (a) the top line from left to right becomes the bottom line from right to left, (b) the second line from the top becomes the next to the last line, also in reverse order, (c) and so on for each screen line. The shortest solution is the best.

Problem #20-2: Numeral Converter

Jim Speers (Niles, MI) suggested this problem. Write a program which will input numbers in English and translate them into numeric form. For example, "eight hundred thirty four thousand five hundred seventeen" would be converted to "834517." Make your program a "barebones" solution with no frills.

Problem #20-3: Decimal Endings

Martin Levinton (New Rochelle, NY) proposed the following challenge. Write a program which prints the integers 1 through 100 and states whether the reciprocal of each number is a repeating or a terminating decimal. If it is a repeating decimal, the program must state the period of repetition. Some examples are in order here. The number 4 has a reciprocal of 1/4 which has a decimal value of 0.25 which is exact (terminating). The number 11 has a reciprocal 1/11 with a repeating decimal representation 0.090909...and a period of 2. Any questions?

Problem #20-4: Hyphen Help

The user inputs a word. The computer displays all reasonable ways of hyphenating the word. For example, if the word is "reciprocal," the output should be

re-ciprocal
reci-procal
recipro-cal

and if the word is "good," the output should be "good." If you submit a solution to this one, please summarize in REM statements or elsewhere the rules you are using.

This month we will discuss readers' solutions to *Commodares* from April. A few people thought that *Problem #19-1: Numerical Columns* was an April Fool's joke, since it cannot be solved in general. The problem was born of necessity since I occasionally need numbered checkoff lists, and using the computer to create them seemed like the thing to do. The problem was to print out N numbers in C columns with all columns being of equal length except for the last one, which might be shorter. If the user enters 5 for the number of columns and 11 for the highest number to be printed, a conflict arises. Clearly there is no arrangement of 11 numbers into 5 columns with the last column shorter than the other four.

Unfortunately, no one sent an analysis of what's wrong with the problem. If the user specifies either one or two columns, there is no conflict for any number of items. With three columns, it is impossible to satisfy the conditions of the problem when N equals 1 or 4. (This assumes that the last column need not contain any items.) Similarly if C equals 4, troublesome values of N are 1, 2, 5, and 6. If C equals 5, N cannot equal 1, 2, 3, 6, 7, 8, or 11. If you can come up with a general explanation for which values of N may not be specified for a given C, let us know. For example, if C is 20, can you predict which values of N won't fit?

Among the simplest solutions to *Problem #19-1* was this one from John K. Lunde (Orono, ME).

```
1 REM
2 REM COMMODARES #20
```

```
3 REM PROBLEM #16-1 : NUMERICAL COLUMNS
4 REM SOLUTION BY JOHN K. LUNDE
5 REM
10 INPUT"COLUMNS";C:INPUT"LAST NUMBER";L
:I%=(L-1)/C+1
20 Y=Y+1:FOR C=Y TO L STEP I%:PRINT RIGH
T$("     "+STR$(C),5);:NEXT:PRINT
30 IF Y<I% THEN 20
```

Another way of rounding a number up to the next larger integer is shown in this solution from John R. Prager (Bay City, MI).

```
1 REM
2 REM COMMODARES #20
3 REM PROBLEM #16-1 : NUMERICAL COLUMNS
4 REM SOLUTION BY JOHN R. PRAGER
5 REM
100 INPUT "NUMBER OF COLUMNS";C%: IF C%<
1 OR C%>5 THEN 100
110 INPUT "ENDING NUMBER";N%: IF N%<1 OR
 N%>1000 THEN 110
120 X=INT(N%/C%+.9):FOR J=1 TO X
130 FOR COL=0 TO C%-1:IF J+COL*X<=N% THE
N PRINT TAB(COL*6) J+COL*X;
140 NEXT:PRINT:NEXT
```

Solutions to this problem and others were received from Jim Speers (Niles, MI), Joey Klein (Chalmette, LA), Willie J. Fitzpatrick, Jr. (Huntsville, AL), Jim Barranti (Canonsburg, PA), Ira Kroll (College Park, MD), J.R. Tomasello (Crosby, TX), Roger Leger (New Westminster, BC), Jacqueline Callaway (Orange Beach, AL), James Borden (Carlisle, PA), Frank Gourley (Kenesaw, NE), Bill Fahber (Bridgeton, NJ), Brano Zidovec (Kitchener, ONT), Kevin Collins (Toledo, OH), Gary Hudach (Youngstown, OH), Grace Ippolitto (Denver, CO), and Richard Gray (Ludlow, VT).

The shortest routine to solve *Problem #16-2: Quick Decimal* was submitted by Brano Zidovec (Kitchener, ONT). He points out that the only problem with his solution is that it does not limit the input to numbers and the letters A through F. Still, the problem requested the *shortest* routine, so here it is:

```
1 REM
2 REM COMMODARES #20
3 REM PROBLEM #16-2 : QUICK DECIMAL
4 REM SOLUTION BY BRANO ZIDOVEC
5 REM
10 INPUT A$:FORI=1 TO LEN(A$):B=ASC(RIGH
T$(A$,I)):C=C+(B-48+7*(B>57))*16^(I-1)
20 NEXT:PRINT C
```

Solutions for this problem came from others not mentioned above including Scott Wilfong (Cedar Rapids, IA), William H. Feyer (Atlanta, GA), Don Wynkoop (Dover, DE), and Garry Hurley Jr. (Williamsport, PA).

The number of correct responses to *Problem #16-3: Time Warp* indicates that many readers know physics as well as computers. The solution below suggested by Bill Fahber (Bridgeton, NJ) is representative of several others.

```
1 REM
2 REM COMMODARES #20
3 REM PROBLEM #16-3 : TIME WARP
4 REM SOLUTION BY BILL FAHBER
5 REM
10 INPUT"++ELOCITY, % OF C";V
20 PRINT"+HAT IS";V*6706166.256"MI/HR"
30 PRINT"OR";V*2997.924562;"KM/SEC"
40 PRINT"+OUR YEAR IS";(1-(V/100)^2)^-.5
;"YEARS."
```

A few embellishments added by other readers are worth noting. John Prager (Bay City, MI) included a statement in his program that the velocity to be entered must be less than 100 percent of the speed of light since "no tachyons are allowed!" R.J. Kaider (Chicago, IL) gave credit in his REM statement to "H. Lorentz, G. Fitzgerald, A. Einstein, et al." James Borden (Carlisle, PA) in his COMAL language solution allowed the user to enter a rocket speed of 100 percent the speed of light. His program responded that a year on the rocket would seem like infinitely many years on earth in such a case. A few readers mentioned that it doesn't matter whether we assume that the earth or the rocket is moving. Time as measured from our system is always observed to be progressing more slowly in the other system. As Jim Speers (Niles, MI) pointed out, "That's why they call it relativity!" Keith Kushner (Brooklyn, NY), Tim Sapp (Mt. Vernon, OH), and Mark Crother (Wilmington, CA), in addition to readers mentioned earlier, sent solutions to this *Commodare.*

If your interest has been whetted by this little problem, two books that I highly recommend are *The Relativity Explosion* by Martin Gardner (Vintage, 1976) and *Mr. Tompkins in Paperback* by George Gamow (Cambridge University Press, 1967). Both books are most enjoyable and informative reading.

Problem #16-4: Common Pairs was a bit of a challenge. The most common approach was to fill a 26 by 26 array with the count of each occurrence of every pair of letters in the given sentence. Fred Ransom (Oxnard, CA) referred to this as a distribution sort. The solution from Jim Speers (Niles, MI) is listed below.

```
1 REM
2 REM COMMODARES #20
3 REM PROBLEM #16-4 : COMMON PAIRS
4 REM SOLUTION BY JIM SPEERS
5 REM
100 DIM M(26,26)
110 PRINT CHR$(147)CHR$(17)"ENTER TEXT.
USE + FOR CONTINUATION":PRINT
115 PRINT"MAXIMUM=255 CHARACTERS":PRINT:
```

WHITE HOUSE COMPUTER

PHONE REBATES! **1-717-322-7700**

P.O. BOX 4025,
WILLIAMSPORT, PA 17701

HOURS: Monday thru Friday 9:00 am till 6:00 pm

VISA and MC Accepted 4%

DISK DRIVES
C-64
1541 .... 185.95
1571 .... 249.95
1572 .... 375.95
MSD Single 229.00
MSD Dual 459.00
Indus GT 239.00
Cardco SCD-1 ... 199.00

INTERFACES
Tymac
Connection/2K ... 62.95
Cardco +G 59.95
Cardco B 37.95
OR - D
GPC w/2K 55.95
Grappler CD 89.95

PRINTER PAPER
Lazor 1000 shts ... 15.95
Lazor 2500 shts ... 29.95

DISKETTES ★
SKC
SS/SD 10.50
SS/DD 13.50
DS/DD 15.50
MAWELL
MD-1 .... 15.95
MD-2 .... 21.95
ELEPHANT
SS/SD 13.50
SS/DD 15.50
DS/DD 19.50

INNOVATIVE CONCEPTS
Flip-N-File
10 .... 3.50
15 .... 8.50
25 .... 18.50
w/lock
50 .... 17.50
50 .... 23.50
ROM Holder ... 17.50

KOALA
Pad Disk 39.95
Pad ROM 45.95

★ PRINTERS ★

C-64
MPS 801 .... 179.95
802 .... 208.95
803 .... 175.95
MCS 801 .... 314.95
DPS 1101 .... 314.95

CITIZEN
MSP-10 325
MSP-15 479
MSP-20 469
MSP-25 629

EPSON
RX80 219
LX80 249
FX80 369
JX80 529
RX100 369
FX100 550

C. ITOH
8510 A 285
8510 BC1 329
8510 BC2 389
8510 BP1 339
8510 SP 389
8510 SR 429
8510 SCP 459
8510 SCR 477
1550 BCD 485
1550 P 445
A10-20P 465
F10-40P/S 875
F10-55 P/S 1059
F10 Tractor 169
F10 Sheet Feeder ... 349

MANNSEMAN TALLY
Spirit 80 249
MT 160L 495
MT 180L 595

STAR MICRONICS
SG10 217
SG15 379
SD10 345
SD15 450
SR10 485
SR15 585
Powertype 309

CARDCO
LQ 1 365
★LQ 3 209

PANASONIC
1090 187
1091 257
1092 389
1093 585
3151 455

LEGEND
880 209
1080 229
1380 269
1385 305

OKIDATA
182 .... 219.95
84 .... 640.95
92 .... 349.95
93 .... 560.95
Imagewriter 425.95

MONITORS
AMDEK
300G- 119
300A 129
310A 145
Color 300 235
Color 500 345
Color 600 399
Color 700 469
Color 710 539

TEKNIKA
MJ-10 229
MJ-22 285

XTRON
Comcolor I 189

ZENITH
122 .... 85.95
123 .... 75.95

BMC
Amber 82.95
Green 79.95

COMMODORE
C-1702 .... 199.95
C-1802 .... 204.95
C-1902 RGB 259.95
C-1901 Monchrome ... 129.95

Dealers Inquiries Invited ★



SOFTWARE ★

BATTERIES INCLUDED
Paper Clipw/Spell Pk ... 69.95
Paper Clip 59.00
Home Pack 32.95
Bus Card II 124.95
B180 Board 99.95

BRODERBUND
Print Shop 29.95
Graphics Library 18.95
B.S. Writer 32.95
EPYX
FAST LOAD 23.95

PRINTER RIBBONS

DUST COVERS
Mitey Mo ... 59.95
Westridge ... 69.95
Compuserve
Starter Kit ... 21.95

VIDEO CASSETTE ★★ RECORDER $295.95

Symphonic VHS/Remote
- 105 Channel Cable Ready
- 14 Day / 1 Program Timer
- 12 Pre-set Channels
- Auto Rewind

Plus much, much more!

CHECK OUR PRICES!

COMPUTERS
C-64 ..... 149.95
C-128 ... 275.95

MODEMS
C-1600 .... 44.95
C-1650 .... 54.95
C-1660 .... 54.95
C-1670 ...... 185.95
300/1200 Band

Cardco
MOD-1 .... 59.95
C-64
MPP-1064 ... 65.95
Tele
Learning 49.95

WE CARRY A FULL LINE OF:
Batteries Inc., Broderbund, Contineal, Hess, Infocom, Innovative Concepts, Koala, LJK, Microbits, Micropose, Microtek, O.S.S., Professional S.W. Scarborough, S.S.I., Synapse, Time Works, Wico.

POLICY: No deposit on COD orders. FREE freight on all prepaid cash orders over $300 in the continental USA. APO and FPO add $5 per hundred. For priority Mail add $8 per hundred. PA Residents add 6% sales tax. Defective products must have Prior RA number. Schools net 15.

Reader Service No. 224

```
S$=""
120 POKE631,34:POKE198,1:INPUTA$:IFRIGHT
$(A$,1)<>"+" THEN 140
130 S$=S$+LEFT$(A$,LEN(A$)-1):PRINT"LENG
TH=";LEN(S$):GOTO120
140 S$=S$+A$
145 FORX=1TOLEN(S$)-1:X$=MID$(S$,X,1):GO
SUB200:A=Z
150 X$=MID$(S$,X+1,1):GOSUB200:B=Z:M(A,B
)=M(A,B)+1:NEXTX
160 PRINTCHR$(147)CHR$(17)S$:PRINT:FORA=
1TO26:FORB=1TO26:IFM(A,B)<2THEN180
170 PRINTCHR$(A+64)CHR$(B+64)"=";M(A,B),
180 NEXTB,A:END
190 PRINTCHR$(A+64)CHR$(B+64)"=";M(A,B),
200 Z=ASC(X$)-64:IFZ<1ORZ>26THENZ=0
210 RETURN
```

Jim's program allows the user to enter the [back arrow] as a continuation character in order for the INPUT statement to handle strings longer than 80 characters. Line 120 POKEs a quotation mark into the keyboard buffer and POKEs a one into the keyboard buffer counter. That allows the user to type commas in the input sentence.

The 27 by 27 array M( ) uses the zero row and column for "garbage collection" such as spaces and non-alphabetic characters. Each letter pair is converted to two numbers which define the element of M( ) to be incremented. For example, the letters "AZ" would be translated to "1,26" in line 200. Matrix element M(1,26) is then incremented in line 150. Line 170 scans through matrix M( ) looking for any elements with a value of 2 or greater, indicating that the pair occurs more than once. The common pairs are printed by line 170.

How many readers know the most common character used in standard text in the English language? The letter "E" you say? Nope. One character is more frequent. How about the frequency of letter pairs or triples? A most impressive analysis of such matters (as well as many others) is contained in *Scientific and Engineering Problem-Solving with the Computer* by William R. Bennett, Jr. (Prentice-Hall, 1976). If you ever run out of things to do on your computer, Dr. Bennett's book will provide a limitless source of ideas.

Here's one final challenge for those of you interested in the problems of letter frequency and cryptography. Modify your solution to *Problem #16-4* so that it can read and analyze text from tape or disk files. Type a large block of text to a file, then calculate the letter frequencies and compare them with other such files. Use your calculated frequencies to decipher encoded text where one letter has been substituted for another. Jg zpv dbo efdjqifs uijt tfoufodf, xsjuf boe ufmm vt ipx zpv eje ju. Happy programming! □

REVIEWS

FELONY!
CBS Software, Inc.
Commodore 64
Disk; $34.95

Last year's *Murder By The Dozen* challenged would-be detectives to solve 12 homicides in the little town of Micropolis. The same design house, BrainBank, has finally delivered this long-awaited sequel.

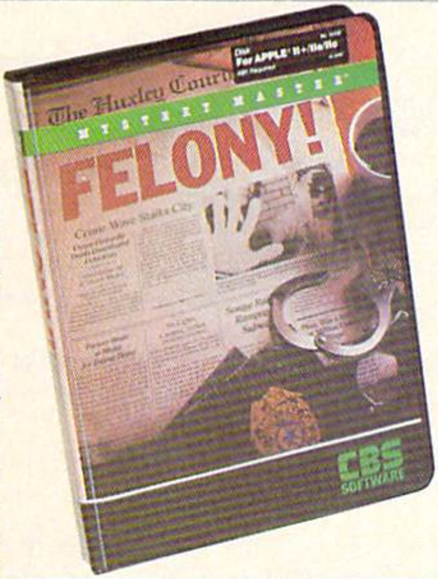
The pressure is mounting on Sean M. McHugh, the police chief of Huxley, Anywhere, U.S.A. The governor wants action on a dozen unsolved cases ranging from the theft of a major archaeological find to a terrorist threat to blow Huxley right off the map.

One to four armchair investigators compete against each other and the time limit to discover the guilty party and unearth enough clues and evidence to help the prosecutor win a conviction when the case goes to trial.

Felony! is a strategy game built around a set of interactive onscreen menus. Each turn, the crime computer offers every detective a choice of seven possible actions. A player can only perform one activity per turn. The precise options which are available depend on the case involved and the sleuth's current whereabouts.

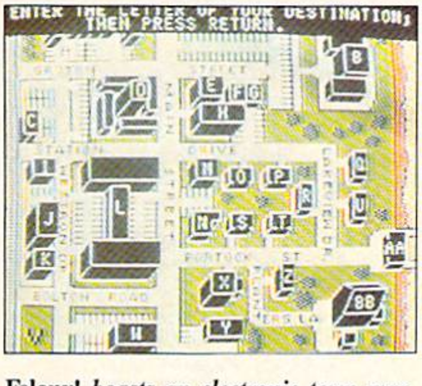
In broad terms, a player can question one of several suspects, examine physical evidence, or move to a different location to pave the way for a new line of inquiry. The police headquarters has some special options found nowhere else in town. These include a records computer with data about known criminals, a laboratory to analyze evidence, and access to your favorite stool pigeon.

The detective enters the number printed next to his or her choice on the menu to select an action. The program responds with a list of several numbers which are keyed to several of the 700 entries printed in the *Felony!* clues book. The player looks up each code number in the clues book and writes down a series of



Felony!*: the long armchair of the law.*
READER SERVICE NO. 176

short phrases which combine to form a message. Sometimes, all that work gets a frustrating "no clue" result, but the investigator usually gets one or more juicy facts that can help catch the perp and solve the crime. A result of "99-99" indicates that the detective is barking up the wrong tree and



Felony! ***boasts an electronic town map, lacking in prequel*** **Murder by the Dozen.**

should head in a new direction.

The clues system is undeniably cumbersome, but it gets the job done. It effectively prevents a player from inadvertently noticing additional clues while receiving information he or she is legitimately entitled to.

Every activity expends precious time. A compu-cop must wrap up the case as quickly as possible to receive a high rating, and many of the crimes must be solved by a deadline. It takes three minutes to question someone, five minutes to examine evidence, and a variable period to travel from place to place.

The program prompts each detective in turn to perform an activity. The crime computer gives any police officer who has accumulated at least 20 minutes on the crime computer clock the chance to advance a solution to the case at hand. To win, an investigator must not only name the criminal but also provide supporting data in the form of motive, evidence, and clues.

A player who wants to charge a suspect looks up the case in the solutions book. By placing a red plastic sheet over the appropriate page, the accuser can read the actual facts of the case. If the solution is correct, that detective wins the game and receives a competence rating from the program. If the theory is wrong, the game proceeds, minus the erring sleuth, until someone does come up with the right answers.

Felony! is certain to please anyone who enjoyed *Murder By The Dozen*. The sequel has improved artwork, an electronic town map, more intriguing situations, multiplayer capability, and a more understandable format for the onscreen menus.

No computer game is perfect, and *Felony!*, too, has defects. The most glaring is that the visuals are still not on the same high level as the gamesystem. BrainBank has clearly made an effort to upgrade this facet of its work, but there is still room for improvement. An illustration for each of Huxley's 28 locations would dress up a program like this considerably.

When all is said and done, it's the quality of the cases which distinguishes a computer mystery game. And the dozen crimes of *Felony!* provide an enjoyable test of the ability to think logically and solve complex problems.

Slik Load

$29.95

Slik Load is a Kartridge for the C-64.
Slik Load is the most reliable, effective and thought out Kartridge of it's kind.
The options include:

- **5 times faster load**
- **Eliminates drive rattle when errors are encountered.**
- **Old and un-new Will restore a basic program**
- **Status key will give you information on device number, bytes free and status of the drive.**
- **Silk Load is also fully compatible with the 1541 Super Rom**





$19.95

DMS-Errors 20, 21, 22, 23, 27 & 29
Format Single Tracks
Read Disk Errors

½ Track Reader-read and select ½ track

½ Track Formatter-Format a disk with ½ tracks. This is where the next protection schemes are coming from.

Drive Mon-Disk Drive assembler/dis-assembler. For your 1541.

The Doc-Disk Doctor that reads code under errors.

Sync Maker-Place a sync mark on **any** track out to 41. Also used for protection

Sync Reader-Check for Sync bits on **any** track out to 41.

Change Drive No.-Changes drive number (7-30).

Disk Logger-Finds starting track sector, start and end addresses.

Disk Match-Compare any two diskettes. Byte for byte.

New Wedge-Easier to use DOS wedge.

ID Check-Check ID's on **any** track.

Unscratch-Restore a scratched file.

View-BAM-Visual display of the free and used sectors on a diskette.

Read/Write Test-1541 performance test.

Repair a Track-Repair a track with checksum errors. Reads code under errors and restores track.

Fast Format-Format a disk in just 10 seconds (with verify!).

This is the only utility of its kind. It even has a 3 min. copy on it.

WAR GAMES AUTODIALER



1-**Auto Dial** will automatically dial a set of numbers you choose.
2-**Review Numbers** will review numbers that were answered by a computer.
3-**Save Numbers** will save numbers where a computer answered.
4-**Hardcopy of Numbers** will print out list of numbers where a computer answered.
5-**LOAD Numbers** will load in numbers to continue where it left off.
6-**Continue** will pick up dialing where it was interrupted.

$29.95

KARTRIDGE KRACKER

NOW you can own this unique and powerful tool which will allow you to dump the contents of 8K and 16K cartridges onto disk! But what's really great is that you can also RUN the cartridges programs without plugging in the cartridge! the KRACKER gets YOU INSIDE the cartridge! Put all your favorites on disk and get rid of the clutter. This package provides your with the software and hardware needed to get started. Program on disk included. (Some cartridges require use of external RAM not included)



$44.95

SOFTWARE PROTECTION HANDBOOK

3rd Edition

THIRD EDITION! NOW AVAILABLE!

If you're **tired** of being harassed by **protected software** and too many **copy** programs, then this is the book for you! This 224 page manual covers the gambit from legalities to protection methods to step-by-step back up procedures. Now you can learn both how to **protect and unprotect** software! The techniques covered include copying cartridges to tape or disk, tape protection, and disk protection. Disk protection covers error no.'s 20, 21, 22, 23, 27 and 29 **plus** single track formatting, header modification, header swapping, half track reading and writing, reading and writing modified bit densities, formatting illegal tracks/sectors, sync writing and **more!** The Third edition **explains,** tells how to **detect** and how to **write** them with included software. Eleven useful utilities and many protection listings! Our disk analysis programs reveal the protection methods used on your originals. A diskette with all software is available for a minimum extra charge. This may not be the only book your should have for the C-64, but it is certainly **the one** book you should not be **without!**

C64 Book only	$19.95 US
Book & Disk of all programs	$29.95 US
Vic 20 book . . . Cart. & Tapes only	$9.95 US

THIS MANUAL DOES NOT CONDONE PIRACY

*SHIPPING: $2.00

TOP SECRET STUFF II

All C128 Compatible

- Split Screens (Horizontally)
- Smooth Scrolling
- Save Ram From Under Roms
- No Drive Rattle On Errors
- Triple Drive Head Speed
- Autoboot Maker
- Koala Screen DUmp
- Display GCR
- Fast Disk Eraser
- Protect Scheme For Your Disks
- Write Protect Disk
- Unwrite Protect Disk
- Mini D.O.S. Wedge
- Fast Diskmatcher
- Data Statement Maker
- Unnew
- 3 Minute Copy
- D.M.S.

If you have TSS#1, You'll Like This One! All on one disk **$19.95**

Bulletin Board

Set up and operate your own bulletin board with one or two disk drives. This one has all the features and you can customize it easily yourself.

1-RUN MEGASOFT-BBS
2-CREATE MEGA FILES
3-ADD TO SYSOP'S CORNER
4-NEW SYSOP'S CORNER
5-READ MESSAGES
6-SCRATCH MESSAGE
7-CYCLE MESSAGES
8-READ SYSOP MESSAGES
9—WRITE OPENING MESSAGE
10-READ LOG
11-CYCLE LOG
12-READ DOWNLOAD FILE
13-SCRATCH DOWNLOAD FILE
14-ADD TO OTHER SYSTEMS
15-CREATE OTHER SYSTEMS



$59.95 BBS

- Leveled Access
- Private Message Base
- Up to 300 Passwords
- Expert Mode
- Open Chalk Board for Highest Level Access

Enclose Cashiers Check, Money Order, or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. Canada orders must be in U.S. Dollars. VISA — MASTER CARD — C.O.D.

Programs for C-64 $2.00 S & H on all orders

Software Submissions Invited

MegaSoft Limited

P.O. Box 1080, Battle Ground, Washington 98604

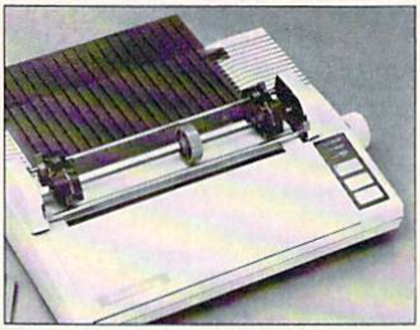
Phone (206) 687-5116 • BBS 687-5205 After Hours Computer to Computer

CBS Software, One Fawcett Place, Greenwich, CT 06836 (phone: 203-622-2500). *—Arnie Katz*

HOMEWRITER 10 PRINTER

Epson America, Inc.
$299.00

The Epson HomeWriter-10 is a friction feed, impact dot-matrix printer that offers 100 cps speed in draft mode and is bidirectional and logic seeking. Additional features include more than 160 different type and style variations, full graphics capability, and an optional tractor feed attachment. Compatibility with most home computers, including the Commodore 64 and VIC 20, is achieved through the use of Printer Interface Cartridges that are available wherever the printer is sold.



NLQ must be seen to be believed.
READER SERVICE NO. 199

When my editor asked me to review the printer, I approached the project with great enthusiasm. I have been in the market for a new printer for some time now and could not pass up the chance to test something new. As an avid computer user I was familiar with the Epson brand name and felt that I was in for a rare treat indeed.

My enthusiasm began to wane, however, shortly after I finished setting up the printer. The setup instructions (a glossy, full color, folding booklet) made no mention of how to access this printer's special features. Nor was there any mention made, aside from a cryptic reference to a mysterious "PIC manual," of the 12 dip switches located on the rear of the printer. A hasty telephone call to the California number listed in the setup booklet as being the Service Division netted me the information that I had, in fact, reached the corporate headquarters and would I please call my "local" service office in Connecticut? As I live in New York City, I was less than thrilled with the prospect of adding yet another long-distance charge to my monthly bill. Feeling that my fee for this review would outweigh the associated telephone charges, I decided that the information I needed justified the cost and placed the call.

I should have quit while I was ahead. My first call to Epson East was answered by a harried sounding woman who, upon hearing my request, promptly placed me on hold. Five minutes later, during which she broke in eleven times to ask if I was being helped, I was disconnected. Repeated calls back netted me 35 minutes of holding time, four more disconnections, and the information that the mysterious "PIC manual" should have been packed in the box containing the interface cartridge. Frustrated and angry, I decided to call once more. This time, using my journalistic credentials, I managed to get through to the advertising department after a twenty minute wait, where a young lady seemed very disappointed that I was not interested in buying ten more printers at a special bulk rate. She was, however, very happy to transfer me to a service technician who cheerfully explained that 50 printers were accidentally shipped without operating manuals and if I would tell him where I purchased my printer he would gladly send me a manual for $3.95! Not having that information at hand, I declined his offer and inquired about a minor glitch with the printer that caused lowercase "o's" to drop below the print line in any mode other than Near Letter Quality. His response was that he had no idea why that happened but he would get back to me. As of this writing, 1 month later, I am still waiting.

The problem of the missing manual was eventually solved by opening up a second interface cartridge package and removing the manual from there. I was able to run the printer through all of its print modes

READER SERVICE INDEX

and find myself quite impressed with it. The Near Letter Quality mode has to be seen to be believed! It even rivals my old Underwood typewriter for the quality of its print.

Though versatile, there are a few problems with this printer. Previously mentioned was the annoying tendency for the lower case "o" to drop slightly below the print line, causing the line to appear ragged and uneven. Another glitch occurred when I used one of the dip switches on the rear of the printer. Supposedly, this switch would cause the printer to place a backslash (/) through the number zero. What actually happens is that the zero is replaced with the graphic symbol you get when you press the Commodore and asterisk keys simultaneously. This is not as big a bug as it could be, though, due to the fact that in all except NLQ mode the difference between the number zero and the capital letter "O" is readily apparent, and most users would not need to make use of the offending switch. The third problem with this printer occurs whenever it is taken offline or runs out of paper. Apparently this causes the printer to send a wait command over the serial bus that causes the suspension of all serial bus operations until the printer is placed back online. I would much rather have the printer return a "Device not ready" error, due to the fact that most decent software will accept such an error and suspend printer operations without crashing. Suspending serial bus operations may lead an inexperienced user to the erroneous conclusion that the computer has "frozen" when, in fact, the printer is simply out of paper.

It should also be noted that the HomeWriter comes from the factory equipped for friction feed only. Users who plan to use pin feed paper, or plan to print more than one page at a time, are advised to purchase the optional tractor feed attachment (list price $39.95). Printing of a long program or document rapidly becomes a chore when you have to stop the printer and realign the paper at the end of each page.

While Epson advertises the HomeWriter-10 as having a print speed of 100 characters per second, this speed is available only in draft mode. Other modes cause the print speed to drop (to as low as 16 cps).

If Epson were to correct the various bugs in this printer, its overall appearance and ease of use would make it a good value at the list price of $299. Commodore owners would then be well advised to consider this as an alternative to the MPS-801. But based on my experience with the Epson Service Department, I cannot recommend this, *or any other*, Epson printer until radical changes in the company's service organization are made.

Epson America, Inc., 2780 Lomita Ave., Torrance, CA 90505 (phone: 213-539-9140). *—B.W. Behling*

MICRO MAIL

Island Digital Concepts
Commodore 64
Disk; $39.95

Micro Mail is a menu-driven program that allows users to easily create and maintain a mailing list of up to 720 friends or business contacts. Once entered, the list can be searched, updated, sorted alphabetically on any field, or printed either in a list format or on individual mailing labels.

Once you've formatted a data disk to hold the information in your mailing list, you simply select the ENTER RECORDS option from the main menu to begin creating your list. Unlike some multipurpose database programs which require you to create your file format from scratch, *Micro Mail* comes with a predesigned mailing list format that is ready to use when you begin. It consists of nine fields including Last Name, First Name, Address, City, State, Zip Code, Phone Number, and a special "Flag" field that can be used to separate friends from business contacts for selection purposes.

Mistakes are no problem to correct, either in the entry mode before you've filed the record, or by selecting the EDIT MODE from the main menu. If you've entered someone's zip code incorrectly, for example, you just enter the number of the zip code

field when prompted and type in the correct information. When you're done making corrections the new version replaces the old. It's also a simple matter to completely remove entries from your list by using the DELETE option. When a record is deleted, its old number is reserved for the next new entry to your mailing list. In this way, the program ensures that all 720 available slots will be appropriated.

The SEARCH/DISPLAY option lets you search your mailing list by filling in the Last Name field for the person you're trying to locate. If you are unsure of the exact spelling, you can still find the entry by supplying at least the first three letters. Unfortunately, searches can only be performed using the Last Name field. If you simply want to browse through the list without a specific entry in mind, you can choose the DISPLAY option and specify a starting record number. Then, using the function keys, you can scroll through the onscreen entries or return to the main menu.

Although specific searches can only be conducted using the Last Name field, SPECIAL LISTS can be created using any of the nine available fields. You could, for example, create a special list of all of the people in your file that have the same specified zip code, or a list of all business contacts if you've used the flag field for that purpose. This special list feature is not without a few limitations, however. First, since you can only use one field to input list criteria, there is no way to create a list of all business contacts who are located in a given state. Also, each time you create a special list, it overwrites the previous one. Thus, it is impossible to maintain two special lists concurrently.

When it comes time to print your mailing list, either in a list format or on individual mailing labels, you may select to print either the entire file or a special list you've created. The entries can even be sorted alphabetically before you print, but here again, you must choose only a single field to sort on. Consequently, you can't sort by last name alphabetically within zip code or by last name and then, when more than one person has the same last name, by first name.

Despite *Micro Mail*'s limitations, its onscreen menus and well-organized documentation make it an excellent choice for C-64 owners who are buying a mailing list program for the first time, or anyone who wants a mailing list program that can be booted and used immediately without first reading through a thick user's manual.

Island Digital Concepts, P.O. Box 1758, Staten Island, NY 10314.

—Bob Guerra

MASTERDISK

Integrated-Software Systems
Commodore 64
Disk; $34.95

When my computer was new, and I wasn't sure what to do with it, a datacassette seemed the perfect choice for program storage. Tapes then seemed equivalent to disks except for speed. Well, that lasted almost two months. First came the upgrade to a disk drive, then the modifications to the drive to make it even faster.

But with speed came the insatiable desire for more programs. Now in a half hour, I can skim through several programs—admiring graphics or just reading title screens—where once I could only load one. The piper demands his pay: I can't find any particular program without a fifteen minute search. Flipping through piles of diskettes has replaced the time my datacassette took to load.

Masterdisk is designed as the cure for this disorganization. It is a cataloging program that automatically reads the directories from your disks, stores them, and allows you to search for the location of individual files. If you have five disks, *Masterdisk* will not save you any time. But if you have fifty, you will soon forget life before cataloging.

Your disk catalog is stored right on the program disk. About 130 disk directories will fit before the program disk is full; 100 is recommended as a cutoff for convenience's sake. For

your second 100 disks, you'll have to use a copy of *Masterdisk* for another catalog. You might then want to reorganize so games are in the first 100, applications are in the second, etc.

You may be surprised, however, at how cataloging will reduce your diskette use. All those partially typed program listings and experimental data files wil be easy to identify. And that one favorite program that's backed up eleven times may be deleted from a couple of disks.

Masterdisk is actually a set of programs that runs from a main menu. There are one- and two-drive cataloging programs. With the one-drive program, you'll have to swap out *Masterdisk* for each data disk you want to catalog. During cataloging, *Masterdisk* will warn you of disks with duplicate ID codes, allow you to change those codes, let you delete files or erase entire disks, and determine the load addresses of machine language programs.

Several different printouts of your catalog will be available. All but one can be sent to the screen if you have no printer. You can print all disk IDs in use; IDs matched to disk names, number of files, and number of blocks free; or all file names sorted alphabetically. If you're trying to fill all your precious disks, you can specify a number of blocks and list all disks with more or fewer blocks free.

Another utility is Disk Changer, which lets you display Block Allocation Maps (BAM), look at the contents of any sector, change disk names without copying the disk, and "trace" through any file sector-by-sector. Best, this utility can restore scratched files as long as they are not overwritten. This single feature can be worth the cost of the program.

The Machine Language Copier utility will move any pure machine language program from one disk to another—no small feat. It can also copy individual program, user, or sequential files that are under 100 blocks long. Last, the Copy 18 utility will read the directories of many copy-protected disks and write unprotected directories onto blank disks, allowing you to make any changes you want before cataloging them. This may be the only method of including your copy-protected programs in the catalog file.

Included with the program disk is a reference card for translating hex, ASCII, binary, and decimal values.

Masterdisk does have two shortcomings. The manual is a minor inconvenience. Its 33 pages are jam-packed, solid text. This visual appearance is reinforced by the contents, which vary from basic information to broad assumptions that you can figure out things like printer codes. Of course, it's impossible to cover all possibilities for all printers in one manual, but beginners will not like parts of these instructions. The manufacturer does suggest in the manual that the befuddled user call with any questions.

The major deficit in this program

is also its biggest asset. Cataloging is automatic. You need not, in fact, can not, enter a program name manually. That name must be on a disk, in the directory sectors, and readable to *Masterdisk.* (There is a way around this: build a dummy disk directory on a separate disk, then use the standard RENAME function through the DOS to specify whatever filename you want to give each file on the dummy disk.) You'll sometimes be forced through convoluted schemes to get copy-protected disk directories into *Masterdisk*'s catalog (but at least you can get them cataloged).

And who cares if a game has three or four separate files. What I want cataloged is the game's name. *Masterdisk* would be improved by a routine to let the user enter disk IDs and file names. Maybe a brief description of each file could also be allowed for those of us who creatively name everything TEST#.

Integrated-Software Systems, P.O. Box 1801, Ames, IA 50010 (phone: 515-233-2992). *—Richard Herring*

FORTH-64

Abacus Software
Commodore 64
Disk; $39.95

FORTH-64 is a computer language that lies, in terms of both speed and complexity, between BASIC and assembly language. The original version of FORTH was developed over a decade ago by Charles Moore. Working in an observatory with a third-generation computer, the IBM 1130, Moore developed what he considered to be a fourth generation language—Fourth. But wait, the 1130 would only accept five-character names; and so FORTH was born.

FORTH found its first home in the science and industry communities. Over the last few years, however, more and more personal computer owners have discovered this language. Actually, to call FORTH a mere language is a misnomer. Leo Brodie, in his outstanding introductory book *Starting FORTH*, describes FORTH as "a high-level language, an assembly language, an operating system, a set of development tools, and a software design philosophy."

As a high-level language FORTH-64 results in programs that run much faster and are much shorter than programs written in BASIC. FORTH programs give you, the programmer, direct access to all the power of the computer. The trade-off is that your FORTH programs will require more organization and be less readable than their BASIC counterparts.

Using as a benchmark the Sieve of Eratosthenes, FORTH executed the program in under a minute, while BASIC required over ten minutes. The FORTH program was about three-fourths the size of BASIC's, and took about three times as long to load.

Compared to assembly language, FORTH is much easier to understand and debug. However, FORTH is not quite as fast; it carries more overhead. A typical FORTH program will run one- to two-thirds slower than a pure assembly language program.

As an operating system, FORTH performs all the usual functions. It assembles, compiles, and interprets code and it handles virtual memory, input, and output. FORTH's operating system is not a separate program, so redundancy and much run-time error checking are done away with. This contributes to FORTH's speed. Since FORTH-64 includes an assembler, those portions of your program that are time critical can easily be debugged in the high-level language, then recoded in assembler.

FORTH-64 is actually a dictionary of defined words, or standard commands, with which you can easily build more commands. In other words, FORTH is extensible. Except for essential words coded in assembler, FORTH's words are programmed in FORTH. The definition of each word must fit on one FORTH screen; 16 lines of 64 characters each. Once any word is defined, it can be executed by just typing its name.

Writing small blocks of code like this requires you to organize your programming, something many of us don't do in BASIC. But, FORTH programming is very different from

other high-level languages. You will find that, in FORTH, even flow charts require a non-traditional approach.

The most immediate and striking difference between FORTH-64 and other languages is that you must code all arithmetic operations in Reverse Polish Notation (RPN), or postfix notation. The equation 12 + (5 * 9) would appear as 5 9 * 12 +. RPN is the result of dealing directly with the stack: that part of memory used by the CPU to store data in a last-in, first-out format.

Many FORTH commands look like familiar BASIC terms or make sense because they are English words. You will find load, list, if, else, and then to be commands similar to BASIC's. Other words, like swap, repeat, loop, copy, and move, will make sense in the context of FORTH's programming environment.

FORTH-64 comes fully equipped with sound and graphics commands to help you use the features of the SID and VIC chips in your C-64. The dictionary includes words for both hi-res and lo-res graphics. Changing screen colors is as easy as typing "BLUE SCREEN BROWN PEN." The use of English words, rather than POKEing numbers into memory locations, is itself enough to convert some people from BASIC to FORTH.

The FORTH Interest Group (FIG) has developed a standard called FIG-FORTH. Abacus' FORTH-64 is based on that standard, but includes three times as many words in its basic vocabulary. Another standard, FORTH 79, is completely included in FORTH-64, and much of the FORTH 83 standard is also included.

Abacus includes a 73-page manual to describe the features of FORTH-64. This manual will not teach you FORTH. Two good books on this language have been authored by Leo Brodie. *Starting FORTH*, from Prentice-Hall, is an introduction to FORTH suitable for either novices seeking to learn about computers or professionals who have just been introduced to FORTH. *Thinking FORTH* discusses some fundamental FORTH concepts, traces the development of a software project, and includes interviews with FORTH programmers.

Abacus Software, P.O. Box 7211, Grand Rapids, MI 49510 (phone: 616-241-5510). *—Richard Herring*

SPREADSHEET

Spinnaker Software
Commodore 64
Disk; $49.95

Spreadsheet packs most of the expected capabilities of a high-powered "what-iffer" into a program that can generate spreadsheets up to 250 rows by 100 columns. The optional 80-column display requires no hardware and is legible on a 1702 monitor, though on a TV screen the characters are more distinguishable in 40-column mode. On top of this, you get limited word processing and database capability plus a slew of unexpected features. The main question to ask is whether or not they will lighten your work load, the object of Spinnaker's line of *Better Working* software.

After using the program to set up, edit, and experiment with "Shay's Pay," a sample spreadsheet recording the millions of dollars I make reviewing software for magazines such as *Ahoy!*, I was impressed with *Spreadsheet*'s comprehensive scope and horsepower. Numbers can be displayed in four ways: integer, decimal, floating decimal, and bar graphs composed of + or − signs. At your command you'll find all the standard functions and a few rare ones: 12 mathematical, 5 range (including COU, which counts the number of numeric entries in a given range), 9 logical operators, and 4 trigonometric functions for the mad scientists in the audience. There are advanced functions such as programmable iteration, Auto Lookup, and an "if" operator that can be employed effectively in conjunction with AND, OR, and NOT when devising formulas. If you're like most people, you're probably still wondering about the extent of the program's database and word

processing capabilities.

When you type text into any cell in a spreadsheet, that cell is referred to as a label and is generally used to name rows and columns: "january, rent, etc." For brief amounts of text, *Spreadsheet* has a "long label" mode in which you can set the maximum width of the text (up to 98 characters) and start typing. The words initially appear in the entry line at the top of the screen. When keyed-in text reaches the limit, it is automatically word-wrapped to the cell below it to produce a left-justified column of text. This facilitates adding an explanatory paragraph below a spreadsheet without much trouble. With the Word command, you set the line width in the same fashion but can also specify right-justification and adjust margins on the sides. Standard cover letters could be saved as separate "spreadsheets" to be customized with new names and addresses when needed. Of course, you may also choose to cobble up a quick letter or memo in a blank section of the current sheet. By setting the range appropriately, you can use partial printing to obtain hard copy of the letter alone. The main benefit is that you don't have to quit the spreadsheet and load a word processor. It's handy for one- or two-pagers, but not for extensive documents.

There is also potential for building databases by filling the rows and columns of a sheet with lists of information instead of formulas to be calculated. One column might hold names, the next addresses, and so on. The ability to search and sort text and numbers makes this feasible, but you don't get a true global search. You must specify in which of four directions from the current cell the search will be conducted. (And the AND, OR, and NOT operators can't be used when searching in this manner). Columns can be sorted alphabetically or numerically, from highest to lowest or lowest to highest.

A practical feature called offset loading allows you to load a file into a blank sheet at a different position from its original one. Parts of files may be saved and loaded by specifying the range, so you can use offset loading to combine lists, graphs, sheets, and text into one document. This feat doesn't qualify *Spreadsheet* as genuine integrated software, but it may be good enough for many people. However, count on investing plenty of time and paper to master the technique: the 146-page manual doesn't provide any info on how to apply these features, forcing you through the rigors or trial and error.

A more offbeat feature is automatic prompting, useful in situations where someone else will be punching numbers into a spreadsheet they didn't create. The person just types "?" and the cursor skips to the first cell that must be filled in. Written by the sheet's architect when setting up the auto-prompts, a line of text pops up in the entry line and tells exactly what kind of data is supposed to be entered in that cell. After the individual hits return, the cursor moves on to the next cell for which an auto-prompt has been set up. There are plenty of other practical features. *Spreadsheet* supports two drives, can send true ASCII, has adjustable baud rates for working with RS232C printers, and furnishes a "draw" feature that fills a range of cells with the same character for drawing visually pleasing borders and frames.

The interface is well-designed, for you don't have to slog through menus or wait for parts of the program to load from disk as with *Multiplan.* Most commands are executed with a control key or / combination: control-f means the next entry is a formula, /L means load a document, and so on. If you change your mind, you can always back out of a command by hitting f7. Cursor control is superb, permitting fast movement to any part of the sheet or a specific cell. Instead of retyping the contents of cells that need to be changed, you can use the insert and delete keys for editing purposes and writeover the existing contents if desired. The most serious drawback is that *Spreadsheet* won't let you delete, insert, or move more than one row, column, or line at a time. Pressing control-r to repeat an insert or delete saves some time, but it would have been more efficient if the program prompted you to state the number of lines or rows to be inserted or deleted.

A "configure" program that is run prior to the actual program helps you get *Spreadsheet* working with a wide range of printers, change display colors, and set other parameters. Options that you might want to alter frequently are accessible from within the spreadsheet. The documentation, though complete and well-indexed, occasionally breaks down with ambiguous phrases that may send you to the phone to dial Spinnaker's technical support number (which is *not* toll-free). A reference card of commands and five screens of RAM-based online help are also available.

Best-suited for small businesses and families, *Spreadsheet* is worth investigating if you can use one or more of its host of unusual features. It's also easy enough for a first-timer to learn and use the fundamental operations.

Spinnaker Software, One Kendall Square, Cambridge, MA 02139 (phone: 617-494-1200).

–Shay Addams

COMMODORE ROOTS

ADDRESSING THE COMMODORE, PART I

THE FIRST OF TWO COLUMNS ON ASSEMBLY-LANGUAGE ADDRESSING

BY MARK ANDREWS

In the first column in this series, I observed that there is a one-to-one correlation between assembly language and machine language—that for every instruction in an assembly language program, there's a corresponding instruction that will mean the same thing when the program has been converted into machine language.

That's the truth, but now that we're further along in our exploration of assembly language, I have to admit that it isn't the whole truth. Actually, there are many assembly language mnemonics (three-letter instructions) that have more than one equivalent instruction in machine language. That means, of course, that there are more instructions in Commodore 64 machine language than in Commodore 64 assembly language.

To understand why this is true—and what it means to the assembly language programmer—it is necessary to have an understanding of *addressing modes,* and how they are used in 6502/6510 assembly language.

If an assembly language program is to work correctly, every statement it contains must be written in a specific format. There are 13 such formats in 6502/6510 assembly language, called addressing modes. Your computer processes each statement it encounters in an assembly language program in accordance with the addressing mode in which the statement is written. Specifically, addressing modes are used to tell a computer how to go about accessing information in its memory, and how to go about storing that information back in its memory once processing is completed.

In this column, we'll examine all 13 of these addressing modes, and observe how they are used in 6502/6510 assembly language. First, though, let's take a look at the eight different ways that one mnemonic—ADC—can be converted into machine language; in other words, the eight *addressing modes* that can be used with the mnemonic ADC:

Assembly Language Addressing Mode	Assembly Language Statement	Machine Language Equivalent	No. of Bytes Used
Immediate	ADC #$03	69 03	2
Zero Page	ADC #$03	65 03	2
Zero Page,X	ADC $03,X	75 03	2
Absolute	ADC $0300	6D 00 03	3
Absolute, Indexed,X	ADC $0300,X	7D 00 03	3
Absolute, Indexed,Y	ADC $0300,Y	79 00 03	3
Indexed, Indirect	ADC ($03,X)	61 03	2
Indirect Indexed	ADC ($03),Y	71 03	2

THE BIG DIFFERENCE

You may notice an odd relationship between the assembly language statements in the second column and the machine language equivalents of those statements listed in the third column. In Column 2 all eight addressing modes use the same mnemonic, but each uses a different operand. In Column 3 the opposite is true. In Column 3, each instruction is different, but only two operands are used: the one-byte operand 03, and the two-byte operand 0300 (listed in column three as 00 03 because of the peculiar low-byte-first convention commonly used in 6502/6510 assembly language).

This is the most important fact illustrated by the above table: in 6502/6510 *machine language*, addressing modes are distinguished from each other by differences in their *op codes.* But in 6510 *assembly language,* the 13 available addressing modes can be identified by differences in their *operands.*

Here's a table that illustrates all 13 addressing modes in Commodore 64 assembly language:

THE 6502/6510 CHIP'S 13 ADDRESSING MODES

Addressing Mode	Format
1. Implicit (Implied)	RTS
2. Accumulator	ASL A (or simply ASL)
3. Immediate	LDA #2
4. Absolute	LDA $02A7
5. Zero Page	STA $FB
6. Relative	BCC LABEL
7. Absolute Indexed,X	LDA $02A7,X
8. Absolute Indexed,Y	LDA $02A7,Y
9. Zero Page,X	LDA $FB,X
10. Zero Page,Y	STX $FB,Y
11. Indexed Indirect	LDA ($FD,X)
12. Indirect Indexed	LDA ($FD),Y
13. Indirect	JMP

ADDNRS.SRC REVISITED

To explain how some of these instructions work, I'll use ADDNRS.SRC, an 8-bit addition routine that you may recall from previous columns. Here's a listing of the ADDNRS program, typed as it would appear if it were created using the *Commodore 64 Macro Assembler.* (If you own a different assembler/editor, such as the *Merlin 64* or *Panther 64,* you should be able to alter the program to meet your assembler's demands without too many problems. If you're a *Merlin 64* owner, it may also help to review last month's column, which explained how to type and run a program using *Merlin.*)

THE ADDNRS SOURCE PROGRAM
(Commodore 64 Assembler Version)

```
10 ;
20 ;ADDNRS.SRC (AN 8-BIT ADDITION PROGR
AM)
30 ;
40  *=$8000
50 ;
60 ADDNRS CLD ;IMPLIED ADDRESS
70  CLC ;IMPLIED ADDRESS
80  LDA #2 ;IMMEDIATE ADDRESS
90  ADC #2 ;IMMEDIATE ADDRESS
100  STA $02A7 ;ABSOLUTE ADDRESS
110  RTS ;IMPLIED ADDRESS
```

Three addressing modes are used in this sample program, and all three are identified in the comments column of the listing. Now let's look at each of the addressing modes that are used in the ADDNRS.SRC program.

IMPLIED (OR IMPLICIT) ADDRESSING

Implied addressing is just what its name—well, implies. In 6502/6510 assembly language, there are a number of mnemonics that never require operands (values that follow an instruction) because they are complete instructions in themselves. So, when you use the implied addressing mode in an assembly language program, all you have to type is a three-letter assembly language instruction; implied addressing does not require (in fact does not allow) the use of an operand.

In the ADDNRS.SRC program, the mnemonics CLD, CLC, and RTS are all implied-addressing instructions.

Op-code mnemonics that can be used in the implicit-addressing mode are BRK, CLC, CLD, CLI, CLV, DEX, DEY, INX, INY, NOP, PHA, PHP, PLA, PLP, RTI, RTS, SEC, SED, SEI, TAX, TAY, TSX, TXA, TXS, and TYA.

IMMEDIATE AND ABSOLUTE ADDRESSING

Immediate addressing and *absolute addressing* are the most common forms of addressing in 6502/6510 assembly language. In the ADDNRS.SRC program, the statements LDA #2 and ADC #2 use immediate addressing. And the absolute addressing mode is employed in the statement STA $02A7.

It's easy to tell the difference between immediate and absolute addressing. In an immediate-addressing statement, a "#" sign always appears in front of the operand. But there is never such a sign before the operand in a statement that uses absolute addressing. In the statement STA $02A7, which uses absolute addressing, a dollar sign does appear before the operand $02A7. But that means that $02A7 is a hexadecimal number; it has nothing to do with the addressing mode of the statement $02A7.

Here's the most important difference between immediate addressing and absolute addressing: in the *immediate* addressing mode, the operand that follows the mnemonic is always interpreted as a *literal number.* But in the *absolute address,* the operand is always interpreted as the address of a *memory location.* So, in the ADDNRS.SRC program, LDA #2 means "load the accumulator with the literal number 2" and ADC #2 means "add the literal number 2 to the content of the accumulator, along with a carry." And the statement STA $02A7 means "store the content of the accumulator in *memory register* $02A7."

Instructions that can be used in the immediate address mode are ADC, AND, CMP, CPX, CPY, EOR, LDA, LDX, LDY, ORA and SBC.

Mnemonics that can be used in the absolute addressing mode are ADC, AND, ASL, BIT, CMP, CPX, CPY, DEC, EOR, INC, JMP, JSR, LDA, LDX, LDY, LSR, ORA, ROL, ROR, SBC, STA, STX, and STY.

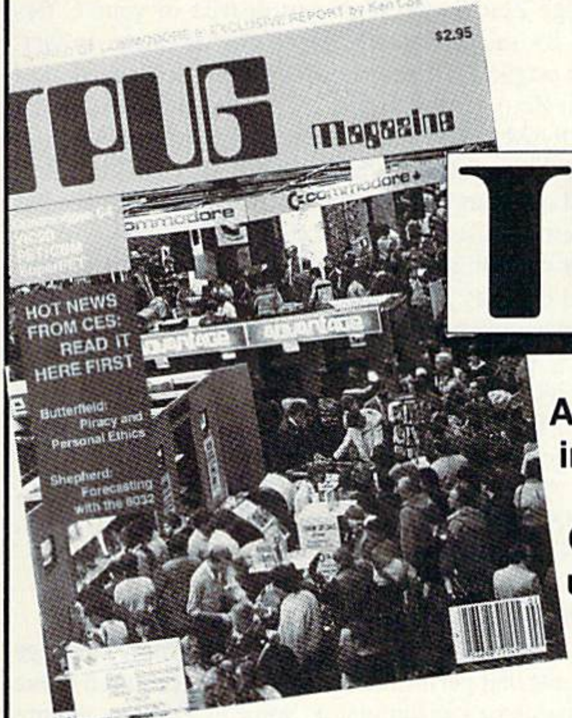
ZERO-PAGE ADDRESSING

Zero-page addressing is used in exactly the same way as absolute addressing. The only difference is that the operand used in a zero-page address always comes from a block of RAM known—logically enough—as Page Zero.

In the Commodore 64, the addresses of the memory locations on Page Zero range from $00 to to $FF. That means that any address on Page Zero can be expressed as a single byte, rather than in the two-byte form needed to express addresses that range from $0100 to $FFFF (addresses that do not lie on Page Zero).

Since Zero-Page addresses can be written so compact-

THE TORONTO PET USERS GROUP INC.



TPUG



A membership in the world's largest Commodore users' group will provide you with:

- 10 issues of TPUG magazine
- Advice from experts like Jim Butterfield and Elizabeth Deal
- Access to a huge library of outstanding public domain software
- An invitation to attend the Annual TPUG conference, May 25 and 26 in Toronto, Canada, with two full days of seminars on all aspects of Commodore computing.

JOIN NOW!

I would like to join TPUG as an associate member at a cost of US$25.00.

Send your cheque, money order or credit card number to:

TPUG Inc.,
1912A Avenue Road,
Toronto, Ontario, Canada
M5M 4A1

Name ______________________________

Address ______________________________

____

☐ Cheque ☐ Money Order ☐ Credit Card __________ Amount in U.S. funds $ ______

Credit Card # __________ Expiry date __________ Signature __________

Type of Computer ______ Disk Drive ______ Printer ______ Modem ______ Other ______

ly, they can save both memory and processing time when used in assembly language programs. In fact, as you will see shortly, some forms of 6502/6510 addressing *require* that their operands be Zero-Page addresses.

It's easy to use Zero-Page addressing in an assembly language program. If you want to use a zero-page address as an operand, you just write it as a one-byte value instead of as a two-byte value. Then, when your assembler converts the program into machine language, it will automatically use the zero-page addressing mode.

Here's how the ADDNRS.SRC program would look if Zero-Page addressing were used instead of absolute addressing in Line 100:

THE ADDNRS SOURCE PROGRAM
(With Zero-Page Addressing in Line 100)

```
10 ;
20 ;ADDNRS.SRC (AN 8-BIT ADDITION PROGR
AM)
30 ;
40  *=$8000
50 ;
60 ADDNRS CLD ;IMPLIED ADDRESS
70  CLC ;IMPLIED ADDRESS
80  LDA #2 ;IMMEDIATE ADDRESS
90  ADC #2 ;IMMEDIATE ADDRESS
100  STA $FB ;ZERO-PAGE ADDRESS
110  RTS ;IMPLIED ADDRESS
```

Since so many advantages are connected with the 256 addresses on Page Zero, it won't surprise you to learn that Page Zero is the high-rent district in your C-64's RAM. It's such a valuable piece of real estate, in fact, that the people who designed your computer claimed most of Page Zero for themselves. Unfortunately, almost all of the RAM space on Page Zero is consumed by your computer's operating system, its built-in BASIC interpreter, and other important utilities, and not much space has been left for user-written programs.

Later on in this series – in a column dedicated specifically to memory management – I'll discuss the memory space available on Page Zero in more detail. For now, the most important fact to remember about Page Zero is that it's used in a special form of addressing known as Zero-Page addressing.

Instructions that can be used with zero-page addressing are ADC, AND, ASL, BIT, CMP, CPX, CPY, DEC, EOR, INC, LDA, LDX, LDY, LSR, ORA, ROL, ROR, SBC, STA, STX, and STY.

ACCUMULATOR ADDRESSING

Accumulator Addressing is used in assembly language statements that perform functions on values stored in the 6510 processor's accumulator. Some assemblers require that the operand "A" be used in statements which employ accumulator addressing. An accumulator-addressing statement that uses the operand "A" looks like this:

```
ASL A
```

Other assemblers, such as the *Merlin 64*, will accept accumulator-addressing statements which do not use an operand. An accumulator-addressing statement written without an "A" operand looks like this:

```
ASL
```

When your computer encounters the instruction ASL in a program, the instruction will shift each bit in the accumulator one bit space to the left, with the leftmost bit (Bit 7) dropping into the carry bit of the processor status (P) register.

Other instructions that can be used in the Accumulator Addressing mode are LSR, ROL, and ROR.

RELATIVE ADDRESSING

Relative addressing is an address mode used for a technique called *conditional branching* – a method for instructing a program to jump to a given routine under certain conditions. In assembly language programming, conditional-branching instructions perform a function similar to that of ON...GOTO instructions in BASIC.

There are eight conditional branching instructions – or relative-address mnemonics – in 6502/6510 assembly language. Each begins with "B", which stands for "branch to."

Examples of the conditional-branching instructions that use relative addressing are BBC (branch to a specified address if the carry flag is clear), BCS (branch to a spe-



cified address if the carry flag is set), BEQ (branch to a specified address if the result of an operation is equal to zero), and BNE (branch to a specified address if the result of an operation is not equal to zero).

All eight instructions will be described later in this series, in a column devoted to looping and branching.

WHAT COMPARISON INSTRUCTIONS DO

The eight comparison mnemonics are often used with three other instructions called *comparison instructions.* Typically, a comparison instruction is used to compare two values with each other, and the conditional branch instruction is then used to determine what should be done if the comparison turns out in a certain way.

The three comparison instructions are: CMP, which means "compare the number in the accumulator with…"; CPX, which means "compare the value in the X register with…", and CPY, which means "compare the value in the Y register with…"

Conditional branching instructions can also follow arithmetic or logical operations, and various kinds of testing of bits and bytes.

Usually, a branch instruction causes a program to branch off to a specified address if certain conditions are met or not met. A branch might be made, for example, if one number is larger than another, if the two numbers are equal, or if a certain operation results in a positive, negative, or zero value.

Here's an example of an assembly language routine that uses conditional branching:

AN ADDITION PROGRAM WITH ERROR-CHECKING

```
10 ;
20 ;ADD8BIT
30 ;
40  *=$8000
50 ;
60 ADD8BIT LDA #0
70  STA $02AA
80 ;
90  CLD
100  CLC
110 ;
120  LDA $02A7
130  ADC $02A8
140  BCS ERROR
150  STA $02A9
160  RTS
170 ERROR LDA #1
180  STA $02AA
190  RTS
```

The above program is an 8-bit addition routine with a simple error-checking utility built in. It adds two 8-bit values, using absolute addressing. If this calculation results in a 16-bit value (a number larger than 255), there will be an overflow error in addition, and the carry bit of the processor status register will be set.

If the carry bit is not set, the sum of the values in $02A7 and $02A8 will be stored in $02A9. If the carry bit is set, however, this condition will be detected in Line 140, and the program will branch to the line labeled ERROR–Line 170.

In this sample program, an error will cause the values in Addresses $02A7 and $02A8 not to be added. Instead, a flag–the number 1–will be loaded into Memory Register $02AA, and the routine will end.

ABSOLUTE INDEXED ADDRESSING

An *indexed address,* like a *relative address,* is calculated by using an offset. But in an indexed address, the offset is determined by the current content of the 6510's X register or Y register.

A statement containing an indexed address can be written using either of these formats:

```
LDA $02A7,X
```

or

```
LDA $02A7,Y
```

HOW ABSOLUTE INDEXED ADDRESSING WORKS

When indexed addressing is used in an assembly language statement, the contents of either the X register or

the Y register (depending upon which index register is being used) are added to the address given in the instruction to determine the final address.

Here's an example of a routine that makes use of indexed addressing. The routine is designed to move byte by byte through a string of ASCII characters, storing the string in a text buffer. When the string has been stored in the buffer, the routine will end.

A DATA-MOVING PROGRAM
(Presented as an Example of Indexed Addressing)

```
10 ;
20 ;DATMOV
30 ;
40 TXTBUF=$02A7
50 EOL=$0D
70 ;
80  *=$8000
90 ;
95  JMP DATMOV
96 ;
100 TEXT .BYTE $54,$41,$4B,$45,$20,$4D,
$45,$20
110  .BYTE $54,$4F,$20,$59,$4F,$55,$52,
20
120  .BYTE $4C,$45,$41,$44,$45,$52,$21,
$0D
130 ;
140 DATMOV
150 ;
160  LDX #0
170 LOOP LDA TEXT,X
180  STA TXTBUF,X
190  CMP #EOL
200  BEQ FINI
210  INX
220  JMP LOOP
230 FINI RTS
250  .END
```


TESTING FOR A CARRIAGE RETURN

When the program begins, we know that the string ends with a carriage return (ASCII $0D), as strings often do in Commodore programs.

As the program proceeds through the string, it tests each character to see whether it is a carriage return. If it isn't, the program moves on to the next character. If it is, that means that there are no more characters in the string, and the routine ends.

ZERO-PAGE,X ADDRESSING

Zero-Page,X addressing is used just like Absolute Indexed,X addressing. However, the address used in the Zero-Page,X addressing mode must (logically enough) be located on Page Zero. Instructions that can be used in the Zero-Page,X addressing mode are ADC, AND, ASL, CMP, DEC, EOR, INC, LDA, LDY, LSR, ORA, ROL, ROR, SBC, STA, and STY.

ZERO-PAGE,Y ADDRESSING

Zero-Page,Y addressing works just like Zero-Page,X addressing, but can be used with only two mnemonics: LDX and STX. If not for the Zero-Page,Y addressing mode, it wouldn't be possible to use absolute indexed addressing with instructions LDX and STX—and that's the only reason this addressing mode exists at all.

INDIRECT ADDRESSING

There are two subcategories of indexed addressing: indexed indirect addressing, and indirect indexed addressing. Both are used primarily to look up data stored in tables.

If you think the names of the two addressing modes are confusing, you're not the first one with that complaint. I never could keep them sorted out myself until I dreamed up a little memory trick to help eliminate the confusion.

Here's the trick: *Indexed indirect addressing*—which has an "X" in the first word of its name—is an addressing mode that makes use of the 6510 chip's X register. *Indirect indexed addressing*—which *doesn't* have an "X" in the first word of its name—uses the 6510's Y register.

In next month's column, we'll look at each of your Commodore's two indirect addressing modes—beginning with *indexed indirect addressing.* □

CADET'S COLUMN

A New Monthly Feature for Beginning Users of the Commodore 64

By Cheryl Peterson

I'm probably not the only person in the world with a complete collection of *Ahoy!* magazines, but I may be the only one who has given them the in-depth attention of a writer searching for an article idea. To sell a piece to a magazine that prides itself on helping Commodore users get the most from their computers, you'd better have a really original idea. I thought I'd never come up with something that hadn't been done before. On the verge of despair and contemplating a two hour ice cream binge, I called the editor with one last desperate query. "No, I've already got Morton working on that." (That Kevelson, he gets all the good assignments, I mentally complain.) "But we've been thinking about starting a beginner's column. Would you be interested?"

Interested? Me? I was delighted. Being a self-appointed apostle of small computers, I always look forward to getting novices really involved with their computers. Sure, it's nice to buy a good software package and learn how to run it. But as far as I'm concerned, you aren't computer literate until you can use some computer language (BASIC, COMAL, PASCAL, or something else) to print a message on the screen, read data off the keyboard, and then do something with the information. Doesn't sound too tough, does it? You might be surprised at how many Apple Macintosh owners will never learn to do it.

Does that mean this column is going to be devoted only to teaching computer language? No! We will touch on several different languages, including BASIC. But, I want this column to help you derive the maximum benefit from the time you spend with your computer. By doing things together, we'll get to know the Commodore 64. Expect me to refer you to your user manual occasionally, because I don't intend to duplicate material found there.

I also won't recommend trying to program everything yourself. Why duplicate the efforts of talented and ingenious individuals who are trying to make a living at something they do well? If you are like me and value your time, you won't want to waste weeks programming your own word processor, spreadsheet, or telecommunications program. I consider a week of my time to be worth at least $100, so if I spend less than that for a good program I feel it's money well spent. Not all programs are worth the price on their label, and where possible I'll try to steer you away from "rotten eggs" and toward "golden eggs." So, sometimes this column will help you find good software values.

One thing new owners should plan at the outset is a computer budget. Before too long, you'll be looking at hordes of software and peripherals you're sure to want. I highly recommend taking a look at your bankbook first. Figure out how much you can afford to spend and then stay within your limit.

It's not a bad idea for all the family members to get in on the discussion of what you really need. You might decide first what peripherals you want to buy and the order in which to get them. If you have only a C-64, there are usually four peripherals that make a good system: a storage device (usually a cassette player or disk drive), a printer, a modem, and an external monitor. The order of purchase varies with the users.

For instance, a family in which students use the Commodore for homework assignments and the parents do a lot of word processing might buy a cassette recorder (about half the cost of a disk drive) and a printer first. Later they might want to upgrade to a disk drive, and then finally buy a monitor (so they can reclaim the TV).

Another family whose members are more comfortable with a typewriter might buy a disk drive and modem first. If one of the parents wants to use the computer to track the stock market, this is an ideal choice. Research using online databases is much faster when the information can be stored on disk. The computer could still be used to research school reports (via online encyclopedias and news services) and then typed on the typewriter. A printer and monitor might be purchased later.

Which peripherals and software to buy should be a family shared decision, if all the members will use the computer. Plan to spend money on disks or cassettes, paper,

software, magazines, and other small items as well.

To be brutally honest, if all you have is the computer (and a storage device) and you don't plan to put any more money into it, you might as well stash it in the closet or sell it, because you aren't likely to get much use out of it. Without magazines to get programs from, media to store programs on, or mass market software to run, a computer is a useless device. And all these things cost money. (Well, we'll pretend "theft" doesn't exist.)

After money, the biggest stumbling block for most new users is programming. Programming isn't as difficult as many people think it is. For small or very user specific applications, it is probably the best way to go.

As a learning process, I would highly recommend entering an *Ahoy!* program that you would really like to use. As you key it in, analyzing what each line is doing as you type it, you'll begin to sense patterns. Most programs written in BASIC and published in *Ahoy!* are accompanied by explanations of how they work and what each routine does. In fact, *Ahoy!* may be one of the best learning sources you'll find.

Motivation is the most important factor in learning BASIC. Pick something you really want to do; play a game, edit fonts, create a new character set, convert numbers from hexadecimal to octal or decimal. Whatever! Then find a program in a book or magazine that does what you want. Enter it in and debug it. By the time you get that program working properly, you should have a much better idea of how to do programming.

One additional piece of advice here. Get your family involved. If you don't have a family, get one! Seriously, working together is a big asset to learning. Entering a program without help means looking back and forth between listing and screen. This can bring on a mean headache by the time you're through. If one person reads the program aloud while the other types, things go much faster and more is retained. Establishing a pattern of working together and sharing the computer will also help you avoid family squabbles and "computer widowhood."

Reading a book on BASIC isn't a bad way to start. You'll have a bit of background to help you understand what a program is doing. But until you actually enter a program and debug it, you probably won't know what all the hoopla is about.

Looking through the back issues, I noticed that the *Rupert Report* column has covered just about every as-

Figure 1:
Topics Covered in *The Rupert Report* and *Creating Your Own Games*

	The Rupert Report	*Creating Your Own Games*
Jan. '84	Cursor Manipulation	—
Feb. '84	String Variables	—
Mar. '84	Computational Logic	—
Apr. '84	Computational Inaccuracies	—
May '84	Arrays	—
June '84	Inputting data: Get, Let, Input	From the Player to the Screen
July '84	Sequential Files	Screen Display
Aug. '84	Number Crunching	Word Games
Sep. '84	Video RAM	Text Adventures
Oct. '84	Programming the Joystick	Text Adventures
Nov. '84	Two-Dimensional Arrays	—
Dec. '84	BASIC Program Structure	Fast Graphics with Custom Characters
Jan. '85	Sprite Basics	Flipping Character Sets
Feb. '85	Sophisticated Sprites	How Games Play
Mar. '85	Disk Operating System	Creating a Gameboard Larger than the Screen
Apr. '85	Getting into the Kernal	Joystick Programming
May '85	Assembly Routines in BASIC	Moving Through Color Memory
June '85	Bit Map Graphics	Automatic Note Entry
July '85	Speeding Pixels	Three-Part Harmonies
Aug. '85	Real-World Simulations	Sound Effects

Figure 2:
Some Commodore 64 Typing Programs

New Improved Mastertype
The Scarborough System
25 N. Broadway
Tarrytown, NY 10591
Game tutor—prevent space ships from shooting you out of the sky by typing correct letters, etc. Newer version contains both QWERTY and DVORAK style keyboards.

Wiztype
Sierra On-Line
Sierra On-Line Building
Coarsegold, CA 93614
Using the Wizard of Id cartoon characters, this program features 6 modes and 20 levels of difficulty. Sprinkled with cartoons, progress testing is done by playing a game. You can design your own tests.

Type Right
Commodore
1200 Wilson Drive
West Chester, PA 19380
Features 17 lessons and 4 games.

MicroType, the Wonderful World of Paws
EduSystems, Inc.
South-Western Publishing
5101 Madison Road
Cincinnati, OH 45227
Tutor program aimed at children, featuring 18 short lessons and a cat that does the teaching.

Keyboard Kadet
Mindscape
3444 Dundee Road
Northbrook, IL 60062
Similar to a standard typing course with onscreen illustrations of hand positions, it also has a shoot'em-up game built in. Features QWERTY and DVORAK keyboards.

9 to 5 Typing
Epyx
1043 Kiel Court
Sunnyvale, CA 94089
Game tutor—game sequences from movie *9 to 5* test your progress. Multiple levels.

Keys to Typing
Batteries Included
186 Queen St. West
Toronto, Ontario
Canada M5V 1Z1
Utilizes .traditional classroom techniques.

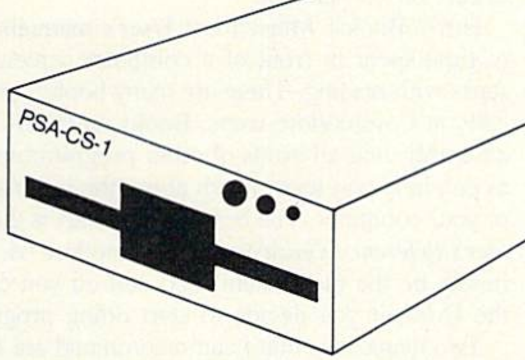
Type Attack
Sirius Software
10364 Rockenham Drive
Sacramento, CA 95827
Shoot-em-up type game.

Typing Tutor—Word Invaders
Academy Software
P.O. Box 6277
San Rafael, CA 94912
Similar to *Mastertype; Space Invaders* with words.

PERIPHERAL SYSTEMS of AMERICA INC.

UNBEATABLE BELIEVE IT!

COMMODORE™
COMPATIBLE
PLUS MORE!



	PSA	Commodore 1541	MSD SD-1	Indus GT
1541 DOS Compatible	YES	YES	YES	YES
Runs all Copy Protected Software	YES	YES	NO	YES
Format Time	16 sec.	90 sec.	18 sec.	18 sec.
@ Save	YES	NO	YES	NO
Data Error Detection & Correction	YES	NO	NO	NO
Switchable Device #	YES	NO	NO	YES
Reset Button	YES	NO	NO	NO
Fast Load & Fast Copy Software	YES	NO	NO	YES
PRICE*	239.99	229.95	299.95	299.95

*Based on Manufacturers List Prices as of April 1, 1985. Dealers may sell for less than suggested list price.

Others may talk, but **no one** can deliver the **features**, the **performance** or the **compatibility** of our remarkable **CS-1 Disk Drive.** Go ahead. Make us prove it! Bring us your toughest test, your trickiest problem — if it works on the 1541™, it will work on the powerful CS-1. But you be the judge . . .

WE'RE BETTER. BELIEVE IT!

The CS-1 is not only **compatible,** it goes one step further. It comes with its own proprietary operating system called **Q-DOS™,** which doesn't have the bugs and quirks that exist in the 1541 DOS. Now, the SAVE function works like it's supposed to. Now, you can reduce read errors with a built-in automatic DATA ERROR CORRECTION capability . . . and much more!

ALL THIS, PLUS AN ADDED BONUS!

Buy a CS-1, and you'll receive the two most popular computer utilities . . . absolutely **FREE.** They include COPY-Q™, the high-speed, single or dual drive copy program that gives you quick and easy back-ups. And Q-LOAD™, the program that speeds up the loading time on most programs. That's almost $100.00 worth of powerful software **FREE** with your new CS-1 Disk Drive!

SO GO AHEAD. PUT US TO THE TEST.

Visit your local computer store today and test drive the new CS-1. You'll like what you see . . . and Seeing is Believing.

Peripheral Systems of America
Products You Can Believe In!

For The Name Of Your Nearest Dealer, Call 214-956-7866 Today!

DEALERS INQUIRIES WELCOME

2526 Manana Road, Suite 209
Dallas, Texas 75220

Reader Service No. 227

pect of BASIC programming imaginable. Frankly, I don't see what else Dale Rupert can find to cover, but I have no doubt he'll come up with something.

In his column on *Creating Your Own Games*, Orson Scott Card has also given many illuminating examples of how to write a good program. In addition to explaining how to create your own routines, he has included program listings for games that you can modify to suit yourself. In some cases, he even tells how to make the changes. (A listing of topics Dale and Orson have covered through August 1985 is in Figure 1.)

For those who haven't got a collection of back issues, I recommend making the investment. There's a lot of good material, including a text editor, disk operating system, sprite manipulator, copy program for sequential and machine language files, a two column directory lister, font editor, and several BASIC programming utilities including an auto line number function, cross referencer, and renumber program.

One very basic skill makes using a computer much easier: touch typing. Four years ago, I used the hunt and peck method. As with many of you, that short personal typing course back in high school taught me what fingers go where, but that was about all. Practice makes proficiency. I now type along at about forty words a minute. Not terribly fast, but I can almost keep up with my thoughts.

Any good typing tutorial programs will help you improve your typing speed. This will make working on the C-64 much more fun. Once you no longer have to direct each finger, things will seem to go much faster. (See Figure 2 for a list of some programs currently available.)

Anyone reading this article already has the most important skill needed to learn to use a computer. Reading! You'd be surprised how many novice computerists believe they have to pay fancy prices for a high school or college course to teach them to use a computer. Nothing could be further from the truth. I'm living proof. I have *never* taken a course in using a computer, yet I am acknowledged by more than a few people as being a relative expert on the subject.

How? Books! Magazines! User's manuals! and a lot of time spent in front of a computer screen. But it all starts with reading. There are many books aimed specifically at Commodore users. Books on BASIC, machine, assembly, and all kinds of other programming languages can help you learn much about the internal workings of your computer. The best place to start is the *Programmer's Reference Guide* to the Commodore 64. It will definitely be the most useful $20 add-on you can buy for the C-64, if you decide to start doing programming.

Two magazines that I can recommend are *Info-64* and *Transactor*. The latter is a Canadian publication that runs a lot of heavy programming material. *Info-64* is a more general magazine featuring software and hardware reviews, new product announcements, and tutorial pieces. (See end of column for subscription information.) Needless to say, if you don't have a subscription to *Ahoy!*, you're missing the boat. (Pun intended; if you missed it send me a self-addressed stamped envelope for an explanation or just fake it and laugh hysterically.)

It doesn't hurt to have someone to talk to when you're really stumped, and if you don't mind waiting a bit for an answer, you could ask me via a letter to *Ahoy!* If you don't have the time to wait, you should check out your local user group. In almost every city in the country, there is at least one computer user group. Frequently there's even a Commodore group. Lists of local user groups can be found periodically in magazines like *Ahoy!* and monthly in *Computer Shopper*. A member may have just the answer you're looking for.

You may have to tolerate a few software pirates accosting you with offers to "trade" programs, but once you tell them you haven't got anything they don't already have, they'll leave you alone. Many user groups also have public domain programs available for next to nothing.

So, you can see that starting to use a computer doesn't have to be a totally overwhelming experience. If you take it slow and watch the road ahead, you'll get there in fine shape. Just don't try to go too far in one day, and watch the speed limit signs. ☐

Info-64
$14.00 per year (6 issues)
Info Publications
Box 2300
Iowa City, IA 52244

Transactor
$15.00 per year (6 issues)
Transactor
277 Lindwood Avenue
Buffalo, NY 14209-9990

SHIP TO SHORE

What Can You Really Do with Your Modem?

By Sheldon Leemon

If you aren't quite sure what computer telecommunications is about, blame Commodore. After all, they joined in the wave of telecomputing hype by advertising that using their computers as inexpensive terminals you could do your homework, your shopping, your banking, your financial planning, send electronical mail, and on and on. With all the publicity surrounding this fast-growing branch of personal computing, it's no wonder some people are confused. Is telecomputing the furtive teenage hacker, spending his bleary-eyed nights trying to break into computer banks full of sensitive information? Is it the all-purpose store, where you can shop without ever leaving your easy chair? Is it an instantaneous mail service that lets you stop playing "telephone tag" and get down to business? Or perhaps a personalized news clipping service that automatically directs stories of interest to your attention? Or a window on all the information stored in the vast libraries of the world?

Interestingly enough, most of the hype surrounding computer communications fails to prominently mention its most common application. What computer users presently spend most of their modem time doing is communicating with other computer users. They do this in a number of ways. First, there are local bulletin board services (BBS's). These services are provided, usually for free, either by groups or by fanatic hobbyists who are willing to dedicate a whole computer system and phone line to the fulltime furtherance of their pastime. One by one, users of these boards call in to read and leave messages, much like a giant version of the corkboard at the drugstore or supermarket. But here, users can often find the solution to a puzzling hardware or software problem. They can also transmit and receive public-domain software via the BBS. The latest in news and software is the essence of the BBS's appeal.

On a larger and more commercial scale are the various subscription information utilities. Services like CompuServe, the Source, and Delphi provide for a fee many of the same functions as the local boards, but on a nationwide basis. By providing local access numbers, these services enable users all over the country to dial into a central database for the price of a local phone call. Here you can find hundreds of messages, and hundreds of programs as well, with new additions appearing each day. Because these information services are run on big mainframe computers, they can handle lots of users at once. This means, among other things, that they can provide a conferencing facility. Such a facility allows a number of users to type messages to one another all at the same time, with everything that one user types appearing on the screens of all the others simultaneously. This enables nationwide "meetings," featuring guest speakers, or on the rowdier, less formal side, "CB simulators."

The user-to-user exchange of ideas and of software makes up the great bulk of the telecomputing done today. By itself, such activity provides ample incentive to buy a modem and get online. In fact, surveys show that computer owners who do get involved with telecomputing are among the most satisfied with personal computing. But what about all those other online activities that we've been promised? Let's take a brief look at some of the available choices.

INFORMATION SERVICES

The promise of up-to-the-minute information accounts for much of the excitement associated with online services. The problems so far are ease of use and cost. For example, many of the big services such as CompuServe and the Source offer news from wire services like UPI and AP. A common way of presenting this news is to serve it up on menus, which give the reader a choice between:

(1) War rages on between Iran and Iraq
(2) Progress stalled in nuclear disarmament talks
(3) Situation tense in Central America
(4) Detroit Tigers win 20th straight game

Obviously, only so many news stories can be presented in this format. By the time you scroll through the menu pages, pick a story, and have it transmitted to your computer, you have expended quite a bit of time, effort, and money, and gained little more information than you would have by tuning into your local all-news radio station on the half hour.

The developing trend is to add to the value of these news services by offering something that a newspaper or newscast can't provide. On the Source, for example, it is possible to check for all UPI or AP national news stories containing the words "Commodore" and "computer." The Executive Information Service on CompuServe offers a custom "clipping service," in which the user registers the topics in which he is interested, and the service automatically routes such stories into his file-box. Another way the information services are beginning to provide useful news services is by offering specialized news not found elsewhere. For example, CompuServe's *On-Line Today* publication offers daily news on developments in the computer and videotex industries.

Of course, news forms just a small part of the information available online. Stock quotations are an example that is commonly cited. While it is true that you can get current information on stock prices online, unless you do a great deal of trading you probably are just as well off calling your broker for a quote. In general, online financial information tends to be costlier than other services. Here again, added services make the difference. The Source has instituted a new program in association with Spear Securities which not only lets you get quotes, but actually trade stocks online. It will also automatically track the value of your portfolio. The Dow Jones News Service offers software that performs sophisticated stock analysis using figures that are downloaded from their service to your computer automatically, but such software is geared mainly toward the IBM PC, and none is available for Commodore products. Besides, it is generally much more expensive to download such data than to acquire it offline from companies such as Value Line, which furnishes the information on disk.

Another frequently touted category of information available online is airline schedules. The Official Airline Guide offers complete scheduling and fare information, and, unlike some proprietary systems used by travel agents, it doesn't "slant" listings towards one particular airline, and so allows the user to find the lowest available fare. While the savings could be substantial if you discover a little-known discount fare, the service itself is not cheap, and at any rate is not one that most people would have occasion to use very often. Recently, CompuServe has added a service which allows you to make reservations directly while online. This enhances the value of the schedule and fare information, since it removes the need to log off and call an agent for reservations. Still, a system where you make the arrangements yourself and pay (by the minute) for the privilege of doing so is hardly a match for the services of a competent travel agent.

One final area of information services that we will look at is the online encyclopedia. Many services offer Grolier's Academic American Encyclopedia online, in various formats. Some versions allow you to search only for titles of articles, while others allow you to actually search for key phrases within articles, and references to other articles. The latter versions can be even more useful than the print version of the encyclopedia. Unfortunately, one encyclopedia is often not enough. When the user can choose between many such references online, the service will be much more useful.

TRANSACTING BUSINESS ONLINE

In this category are services such as online shopping, travel, and stock brokerage. While the potential for such services is great, they are still for the most part in their infancy. A good example is online banking. Many of the large national banks like the Bank of America have started intensive pilot programs for online banking, allowing users to check their bank balances, make automatic transfers from their checking account to pay their Visa bills and charge accounts at large chain stores like Sears and make periodic payments (such as mortgage and car) automatically every month. Still, such programs really don't have enough features to offer yet to offset the service charge, which usually runs in the neighborhood of ten dollars a month. While it can replace much of the checkwriting that an individual does, most people don't write all that many checks. And it cannot help at all with the kinds of cash transactions that most people conduct at banks. In the future, perhaps the banks will offer additional services, such as tracking all of your income and expenditures automatically in the same way that personal finance software does. As for now, however, their online services are of limited interest.

Much the same can be said of online shopping. Almost every major service offers some form of online shopping, and the variety of goods and services available online is staggering. Nonetheless, the promised convenience just is not there. Most systems are fairly clumsy to use, and offer no greater discounts than can be gotten either through local discount houses or mail-order houses with WATS lines. While such systems could be expected to attract the same sort of shoppers who buy through catalogs, here the catalog has no pictures, and you pay to read it. So except for real bargain hunters, people who live far from heavily populated areas, or shoppers look-

ing for rare items that are difficult to locate, these services are of limited usefulness.

ENTERTAINMENT

On most systems, that translates out to games. The major general-interest information services all offer the traditional computer games like chess, *Hunt the Wumpus, Adventure,* etc. But most Commodore users will find much better games available at their local software stores for a lot less money. The one area in which these services can offer something that stand-alone software can't is in multiplayer games, in which many users participate at once. The only information service which really takes advantage of this facet of computer entertainment is CompuServe, which offers multiplayer space-war and fantasy games. These games are among the system's most popular features.

Does all of this mean that telecomputing is just a worthless scam? Of course not. While it may not be the ultimate activity it is sometimes made out to be, it is still one of the fastest growing and most fascinating aspects of personal computing. The real message that I hope to convey is that though there are a number of different aspects to telecomputing, not every one is right for everybody. The new user should not feel compelled to try every service offered all at once, and then become frustrated because it hasn't whitened his teeth or improved his love life. If you are a beginner, you will find that the local bulletin board or special interest group on CompuServe or Delphi will provide lots of exciting information and (almost) free software. As your experience in telecomputing expands, and as the various services mature, you will surely find other areas that will be of interest to you, and you will be ready for them. There is such a wide variety of online services available that you will be able to tailor your telecomputing activity to suit your needs.

As for the more experienced readers, do you agree or disagree with this column? Is there any special telecomputing activity you especially enjoy that you feel I've slighted, or left out entirely? Keep in touch.

Sheldon Leemon
CompuServe ID 72705,1355
Source BBX878

SMALL THINGS CONSIDERED

Our congratulations to the following Commodore users, winners of subscriptions to *Ahoy!* courtesy of New York's Small Things Considered radio show (heard weeknights 5-8 and Saturdays 6-8 on WNYC AM83):

Wayne Notife (Brooklyn, NY) **John Moscow (Bronx, NY)** **Amy Russo (New York, NY)**

S.O.S.

By David Barron, Morton Kevelson, and Michael Davila

Enclosed is the program that I typed right out of the *Simons' BASIC* instruction booklet. When I run it on either my 64 or my SX-64, it displays the sprites but it eats up the program too. What gives? —*Michael Daniels*
CAFB, SC

Michael,

Simons' BASIC *stores the sprite data in two areas of memory. The first extends from 2048 through 4095. The second is from 8192 through 16383. See the DESIGN command in the manual to learn how you may choose the storage location for sprites. The program you sent, from chapter eight in the* Simons' BASIC *manual, uses the first of these areas, which happens to coincide with the start of BASIC's program storage. The following will move the start of BASIC up to 4096:*

```
POKE44,16:POKE4096,0:NEW
```

The first POKE changes the address for the start of BASIC. The second POKE puts a zero in the first location. This is required by the BASIC interpreter. These POKEs will reserve two kilobytes for sprite data storage, while protecting your program from being overwritten by the sprite routines in Simons' BASIC. *These commands must be executed before LOADing the program. The penalty, of course, is the loss of two kilobytes from BASIC memory.*

I bought a 64 last August and right now am getting into hi-res (bit mapped) graphics. I'd like to know if the 64 is capable of printing text and display bit mapped graphics at the same time. —*Orlando Gonzalez*
Bronx, NY

Orlando,

It is not possible to mix the text and graphics (bit map) modes on the 64. One simple way around this is to write a routine that will place text on the bit map by reading the proper codes from the character generator ROM. This is simple to do, and will display characters identical to the standard character set on the 64.

Several months ago I bought a Sears TV/Monitor (catalog #57H4084C) for use with my C-64. It has three operating modes: television, monitor (audio/video using RCA input jacks), and RGB monitor (with RGB style plug). I used the five pin, two plug monitor cable to connect my 64 to the monitor.

This arrangement worked fine for playing games, but problems arose when I started using a terminal emulator that uses 80 columns on the 64. It's extremely hard to decipher characters positioned close together in this 80-column mode. I thought this was because of the 64's limited graphics, but when a co-worker ran the same terminal emulator on his C-64 and 1702 monitor, the difference in clarity was astounding.

The manufacturer states that the TV/Monitor supports 80 columns in RGB mode, but says nothing about the audio/video mode. Is there any way to achieve the resolution of the Commodore monitor on my monitor? Is there any way to convert the Commodore audio/video signal to an RGB signal? —*Sean Flynn*
West Simsbury, CT

Sean,

Unfortunately, you are presently using the best means available. Even though your monitor has provisions for an RGB input, the C-64 does not have an RGB output. The 8-pin cable provides a separation of the video signal into its chroma and luminance components. This can only be taken advantage of by Commodore monitors to provide greater clarity. There is no feeble way to split a standard video signal into its RGB components.

I have just purchased my fourth issue of *Ahoy!* THANK YOU! Your magazine is the best source of info for me and my C-64. But I have a problem. I like to type in and save the programs you list, and it would be simpler to do so if I could use the *Bug Repellent* program. When I run it all I get is "? QUANTITY ERROR IN 5000". Please straighten me out.
—*Larry McCann*
Cincinnati, OH

Dear Crooked,

The problem is not being caused by line 5000, but by the data statements in your program. What most likely caused the error is that you typed a number larger than 255 within the data lines. I would suggest checking your data statements to correct the problem.

PROGRAM LISTINGS

Attention new *Ahoy!* readers! You must read the following information very carefully prior to typing in programs listed in *Ahoy!* Certain Commodore characters, commands, and strings of characters and commands will appear in a special format. Follow the instructions and listing guide on this page.

On the following pages you'll find several programs that you can enter on your Commodore computer. But before doing so, read this entire page carefully.

To insure clear reproductions, *Ahoy!*'s program listings are generated on a daisy wheel printer, incapable of printing the commands and graphic characters used in Commodore programs. These are therefore represented by various codes enclosed in brackets [ ]. For example: the SHIFT CLR/HOME command is represented onscreen by a heart. The code we use in our listings is [CLEAR]. The chart below lists all such codes which you'll encounter in our listings, except for one other special case.

The other special case is the COMMODORE and SHIFT characters. On the front of most keys are two symbols. The symbol on the left is obtained by pressing that key while holding down the COMMODORE key; the symbol on the right, by pressing that key while holding down the SHIFT key. COMMODORE and SHIFT characters are represented in our listings by a lower-case "s" or "c" followed by the symbol of the key you must hit. COMMODORE J, for example, is represented by [c J], and SHIFT J by [s J].

Additionally, any character that occurs more than two times in a row will be displayed by a coded listing. For example, [3 "[LEFT]"] would be 3 CuRSoR left commands in a row, [5 "[s EP]"] would be 5 SHIFTed English Pounds, and so on. Multiple blank spaces will be noted in similar fashion: e.g., 22 spaces as [22 " "].

Sometimes you'll find a program line that's too long for the computer to accept (C-64 lines are a maximum of 80 characters, or 2 screen lines long; VIC 20 lines, a maximum of 88 characters, or 4 screen lines). To enter these lines, refer to the *BASIC Command Abbreviations Appendix* in your User Manual.

On the next page you'll find our *Bug Repellent* programs for the VIC 20 and C-64. The version appropriate for your machine will help you proofread our programs after you type them. (Please note: the *Bug Repellent* line codes that follow each program line, in the whited-out area, should *not* be typed in. See the instructions preceding each program.)

Also on the following page you will find *Flankspeed*, our ML entry program, and instructions on its use. □

Call *Ahoy!* at 212-239-0855 with any problems.

When You See	It Means	You Type		You Will See
[CLEAR]	Screen Clear	SHIFT	CLR/HOME	
[HOME]	Home		CLR/HOME	
[UP]	Cursor Up	SHIFT	↑ CRSR ↓	
[DOWN]	Cursor Down		↑ CRSR ↓	
[LEFT]	Cursor Left	SHIFT	←CRSR→	
[RIGHT]	Cursor Right		←CRSR→	
[SS]	Shifted Space	SHIFT	Space	
[INSERT]	Insert	SHIFT	INST/DEL	
[DEL]	Delete		INST/DEL	
[RVSON]	Reverse On	CNTRL	9	
[RVSOFF]	Reverse Off	CNTRL	0	
[UPARROW]	Up Arrow		↑	
[BACKARROW]	Back Arrow		←	
[PI]	PI		π	
[EP]	English Pound		£	

When You See	It Means	You Type		You Will See
[BLACK]	Black	CNTRL	1	
[WHITE]	White	CNTRL	2	
[RED]	Red	CNTRL	3	
[CYAN]	Cyan	CNTRL	4	
[PURPLE]	Purple	CNTRL	5	
[GREEN]	Green	CNTRL	6	
[BLUE]	Blue	CNTRL	7	
[YELLOW]	Yellow	CNTRL	8	
[F1]	Function 1		F1	
[F2]	Function 2	SHIFT	F1	
[F3]	Function 3		F3	
[F4]	Function 4	SHIFT	F3	
[F5]	Function 5		F5	
[F6]	Function 6	SHIFT	F5	
[F7]	Function 7		F7	
[F8]	Function 8	SHIFT	F7	

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** This and the preceding explain these codes and provide other essential information on entering **Ahoy!** programs. Read these pages **before** entering programs.

BUG REPELLENT

This program will let you debug any *Ahoy!* program. Follow instructions for VIC 20 (cassette or disk) or C-64.

VIC 20 VERSION
By Michael Kleinert and David Barron

For cassette: type in and save the *Bug Repellent* program, then type RUN 63000[RETURN]SYS 828[RETURN]. If you typed the program properly, it will generate a set of two-letter line codes that will match those listed to the right of the respective program lines.

Once you've got a working *Bug Repellent*, type in the program you wish to check. Save it and type the RUN and SYS commands listed above once again, then compare the line codes generated to those listed in the magazine. If you spot a discrepancy, a typing error exists in that line. Important: you must use exactly the same spacing as the program in the magazine. Due to memory limitations on the VIC, the VIC *Bug Repellent* will register an error if your spacing varies from what's printed.

You may type SYS 828 as many times as you wish, but if you use the cassette for anything, type RUN 63000 to restore the *Repellent.*

When your program has been disinfected you may delete all lines from 63000 on. (Be sure the program you type doesn't include lines above 63000!)

For disk: enter *Bug Repellent*, save it, and type RUN:NEW [RETURN]. Type in the program you wish to check, then SYS 828.

To pause the line codes listing, press SHIFT.

To send the list to the printer type OPEN 4,4:CMD 4:SYS 828[RETURN]. When the cursor comes back, type PRINT#4:CLOSE 4[RETURN].

```
•63000 FORX=828TO1023:READY:POKEX,Y:NEXT:END    AC
•63001 DATA169,0,133,63,133,64,165,43,133,251   JL
•63002 DATA165,44,133,252,160,0,132,254,32,228 DF
•63003 DATA3,234,177,251,208,3,76,208,3,230     OE
•63004 DATA251,208,2,230,252,169,244,160,3,32   OH
•63005 DATA30,203,160,0,177,251,170,230,251,20
 8                                              KO
•63006 DATA2,230,252,177,251,32,205,221,169,58 JJ
•63007 DATA32,210,255,169,0,133,253,230,254,32 OK
•63008 DATA228,3,234,165,253,160,0,170,177,251 LG
•63009 DATA201,32,240,6,138,113,251,69,254,170 BP
•63010 DATA138,133,253,177,251,208,226,165,253
 ,41                                            DD
•63011 DATA240,74,74,74,74,24,105,65,32,210     EK
•63012 DATA255,165,253,41,15,24,105,65,32,210   FO
•63013 DATA255,169,13,32,210,255,173,141,2,41   PK
•63014 DATA1,208,249,230,63,208,2,230,64,230    CB
•63015 DATA251,208,2,230,252,76,74,3,169,236    KH
•63016 DATA160,3,32,30,203,166,63,165,64,32     DP
•63017 DATA205,221,169,13,32,210,255,96,230,25
 1                                              EL
•63018 DATA208,2,230,252,96,0,76,73,78,69       OI
•63019 DATA83,58,32,0,76,73,78,69,32,35         FG
•63020 DATA32,0,0,0,0,0                         LE
```

C-64 VERSION
By Michael Kleinert and David Barron

Type in, SAVE, and RUN the *Bug Repellent*. Type NEW, then type in or LOAD the *Ahoy!* program you wish to check. When that's done, SAVE your program (don't RUN it!) and type SYS 49152 [RETURN].

To pause the listing depress and hold the SHIFT key.

Compare the codes your machine generates to the codes listed to the right of the respective program lines. If you spot a difference, an error exists in that line. Jot down the number of lines where contradictions occur, LIST each line, spot the errors, and correct them.

```
•5000 FORX=49152TO49488:READY:POKEX,Y:NEXT:END GJ
•5001 DATA32,161,192,165,43,133,251,165,44,133 DL
•5002 DATA252,160,0,132,254,32,37,193,234,177  DB
•5003 DATA251,208,3,76,138,192,230,251,208,2   OF
•5004 DATA230,252,76,43,192,76,73,78,69,32     KN
•5005 DATA35,32,0,169,35,160,192,32,30,171     CA
•5006 DATA160,0,177,251,170,230,251,208,2,230  CE
•5007 DATA252,177,251,32,205,189,169,58,32,210 JE
•5008 DATA255,169,0,133,253,230,254,32,37,193  CL
•5009 DATA234,165,253,160,0,76,13,193,133,253  NB
•5010 DATA177,251,208,237,165,253,41,240,74,74 MB
•5011 DATA74,74,24,105,65,32,210,255,165,253   EP
•5012 DATA41,15,24,105,65,32,210,255,169,13    GH
•5013 DATA32,220,192,230,63,208,2,230,64,230   AN
•5014 DATA251,208,2,230,252,76,11,192,169,153  NG
•5015 DATA160,192,32,30,171,166,63,165,64,76   BF
•5016 DATA231,192,96,76,73,78,69,83,58,32      EP
•5017 DATA0,169,247,160,192,32,30,171,169,3    PJ
•5018 DATA133,254,32,228,255,201,83,240,6,201  FK
•5019 DATA80,208,245,230,254,32,210,255,169,4  FL
•5020 DATA166,254,160,255,32,186,255,169,0,133 CL
•5021 DATA63,133,64,133,2,32,189,255,32,192    GC
•5022 DATA255,166,254,32,201,255,76,73,193,96  NN
•5023 DATA32,210,255,173,141,2,41,1,208,249    NH
•5024 DATA96,32,205,189,169,13,32,210,255,32   IM
•5025 DATA204,255,169,4,76,195,255,147,83,67   KC
•5026 DATA82,69,69,78,32,79,82,32,80,82        DC
•5027 DATA73,78,84,69,82,32,63,32,0,76         ML
•5028 DATA44,193,234,177,251,201,32,240,6,138  GN
•5029 DATA113,251,69,254,170,138,76,88,192,0   JK
•5030 DATA0,0,0,230,251,208,2,230,252,96       NA
•5031 DATA170,177,251,201,34,208,6,165,2,73    DM
•5032 DATA255,133,2,165,2,208,218,177,251,201  JA
•5033 DATA32,208,212,198,254,76,29,193,0,169   FM
•5034 DATA13,76,210,255,0,0,0                  PA
```

FLANKSPEED FOR THE C-64
By Gordon F. Wheat

Flankspeed will allow you to enter machine language *Ahoy!* programs without any mistakes. Once you have typed the program in, save it for future use. While entering an ML program with *Flankspeed* there is no need to enter spaces or hit the carriage return. This is all done automatically. If you make an error in a line a bell will ring and you will be asked to enter it again. To LOAD in a program Saved with *Flankspeed* use LOAD "name",1,1 for tape, or LOAD "name",8,1 for disk. The function keys may be used after the starting and ending addresses have been entered.

f1 – SAVEs what you have entered so far.

f3 – LOADs in a program worked on previously.

f5 – To continue on a line you stopped on after LOADing in the previously saved work.

f7 – Scans through the program to locate a particular line, or to find out where you stopped the last time you entered the program. f7 temporarily freezes the output as well.

```
•5 POKE53280,12:POKE53281,11                    LL
•6 PRINT"[CLEAR][c 8][RVSON][15" "]FLANKSPEED[
 15" "]";                                       ED
•10 PRINT"[RVSON][5" "]MISTAKEPROOF ML ENTRY P
 ROGRAM[6" "]"                                  MC
•15 PRINT"[RVSON][9" "]CREATED BY G. F. WHEAT[
 9" "]"                                         DM
•20 PRINT"[RVSON][3" "]COPR. 1984, ION INTERNA
```

```
TIONAL INC.[3" "]"                              DH
30 FORA=54272TO54296:POKEA,0:NEXT               IM
40 POKE54272,4:POKE54273,48:POKE54277,0:POKE5
4278,249:POKE54296,15                           NH
70 FORA=680TO699:READB:POKEA,B:NEXT             KO
75 DATA169,251,166,253,164,254,32,216,255,96    HJ
76 DATA169,0,166,251,164,252,32,213,255,96      JB
80 B$="STARTING ADDRESS IN HEX":GOSUB2010:AD=
B:SR=B                                          HC
85 GOSUB2520:IFB=0THEN80                        FO
86 POKE251,T(4)+T(3)*16:POKE252,T(2)+T(1)*16    KE
90 B$="ENDING ADDRESS IN HEX":GOSUB2010:EN=B    IF
95 GOSUB2510:IFB=0THEN80                        FP
96 POKE254,T(2)+T(1)*16:B=T(4)+1+T(3)*16        MN
97 IFB>255THENB=B-255:POKE254,PEEK(254)+1       GE
98 POKE253,B:PRINT                              HN
100 REM  GET HEX LINE                           IL
110 GOSUB3010:PRINT": [c P][LEFT]";:FORA=0TO8   FG
120 FORB=0TO1:GOTO210                           MD
125 NEXTB                                       ME
130 A%(A)=T(1)+T(0)*16:IFAD+A-1=ENTHEN310       LH
135 PRINT" [c P][LEFT]";                        IK
140 NEXTA:T=AD-(INT(AD/256)*256):PRINT" "       PD
150 FORA=0TO7:T=T+A%(A):IFT>255THENT=T-255      LK
160 NEXT                                        IA
170 IFA%(8)<>TTHENGOSUB1010:GOTO110             FK
180 FORA=0TO7:POKEAD+A,A%(A):NEXT:AD=AD+8:GOT
O110                                            MN
200 REM    GET HEX INPUT                        AB
210 GETA$:IFA$=""THEN210                        HO
211 IFA$=CHR$(20)THEN270                        GC
212 IFA$=CHR$(133)THEN4000                      MD
213 IFA$=CHR$(134)THEN4100                      KF
214 IFA$=CHR$(135)THENPRINT" ":GOTO4500         GE
215 IFA$=CHR$(136)THENPRINT" ":GOTO4700         BJ
220 IFA$>"@"ANDA$<"G"THENT(B)=ASC(A$)-55:GOTO
250                                             GM
230 IFA$>"/"ANDA$<":"THENT(B)=ASC(A$)-48:GOTO
250                                             LE
240 GOSUB1100:GOTO210                           LL
250 PRINTA$"[c P][LEFT]";                       OA
260 GOTO125                                     CG
270 IFA>0THEN280                                OP
272 A=-1:IFB=1THEN290                           OB
274 GOTO140                                     CJ
280 IFB=0THENPRINTCHR$(20);CHR$(20);:A=A-1      HG
285 A=A-1                                       BE
290 PRINTCHR$(20);:GOTO140                      KH
300 REM LAST LINE                               AD
310 PRINT" ":T=AD-(INT(AD/256)*256)             GJ
320 FORB=0TOA-1:T=T+A%(B):IFT>255THENT=T-255    PL
330 NEXT                                        IA
340 IFA%(A)<>TTHENGOSUB1010:GOTO110             KF
350 FORB=0TOA-1:POKEAD+B,A%(B):NEXT             HN
360 PRINT:PRINT"YOU ARE FINISHED!":GOTO4000     ON
1000 REM BELL AND ERROR MESSAGES                FL
1010 PRINT:PRINT"LINE ENTERED INCORRECTLY":PR
INT:GOTO1100                                    DH
1020 PRINT:PRINT"INPUT A 4 DIGIT HEX VALUE!":
GOTO1100                                        JA
1030 PRINT:PRINT"ENDING IS LESS THAN STARTING
!":B=0:GOTO1100                                 HD
1040 PRINT:PRINT"ADDRESS NOT WITHIN SPECIFIED
RANGE!":B=0:GOTO1100                            AG
1050 PRINT:PRINT"NOT ZERO PAGE OR ROM!":B=0:G
OTO1100                                         KN
1060 PRINT"?ERROR IN SAVE":GOTO1100             EI
1070 PRINT"?ERROR IN LOAD":GOTO1100             GL
1080 PRINT:PRINT:PRINT"END OF ML AREA":PRINT    PG
1100 POKE54276,17:POKE54276,16:RETURN           BH
1200 OPEN15,8,15:INPUT#15,A,A$:CLOSE15:PRINTA
$:RETURN                                        IM
2000 REM GET FOUR DIGIT HEX                     PC
2010 PRINT:PRINTB$;:INPUTT$                     GM
2020 IFLEN(T$)<>4THENGOSUB1020:GOTO2010         II
2040 FORA=1TO4:A$=MID$(T$,A,1):GOSUB2060:IFT(
A)=16THENGOSUB1020:GOTO2010                     AD
2050 NEXT:B=(T(1)*4096)+(T(2)*256)+(T(3)*16)+
T(4):RETURN                                     GF
2060 IFA$>"@"ANDA$<"G"THENT(A)=ASC(A$)-55:RET
URN                                             EH
2070 IFA$>"/"ANDA$<":"THENT(A)=ASC(A$)-48:RET
URN                                             KP
2080 T(A)=16:RETURN                             NP
2500 REM ADRESS CHECK                           LI
2510 IFAD>ENTHEN1030                            MI
2515 IFB<SRORB>ENTHEN1040                       MG
2520 IFB<256OR(B>40960ANDB<49152)ORB>53247THE
N1050                                           MI
2530 RETURN                                     IM
3000 REM ADDRESS TO HEX                         EB
3010 AC=AD:A=4096:GOSUB3070                     HG
3020 A=256:GOSUB3070                            CE
3030 A=16:GOSUB3070                             PN
3040 A=1:GOSUB3070                              MJ
3060 RETURN                                     IM
3070 T=INT(AC/A):IFT>9THENA$=CHR$(T+55):GOTO3
090                                             CJ
3080 A$=CHR$(T+48)                              JP
3090 PRINTA$;:AC=AC-A*T:RETURN                  AC
4000 A$="**SAVE**":GOSUB4200                    AI
4050 OPEN1,T,1,A$:SYS680:CLOSE1                 LH
4060 IFST=0THENEND                              EO
4070 GOSUB1060:IFT=8THENGOSUB1200               FJ
4080 GOTO4000                                   FF
4100 A$="**LOAD**":GOSUB4200                    AB
4150 OPEN1,T,0,A$:SYS690:CLOSE1                 MF
4160 IFST=64THEN110                             JH
4170 GOSUB1070:IFT=8THENGOSUB1200               CM
4180 GOTO4100                                   FO
4200 PRINT" ":PRINTTAB(14)A$                    FG
4210 PRINT:A$="":INPUT"FILENAME";A$             OM
4215 IFA$=""THEN4210                            GF
4220 PRINT:PRINT"TAPE OR DISK?":PRINT           DF
4230 GETB$:T=1:IFB$="D"THENT=8:A$="@0:"+A$:RE
TURN                                            IG
4240 IFB$<>"T"THEN4230                          FN
4250 RETURN                                     IM
4500 B$="CONTINUE FROM ADDRESS":GOSUB2010:AD=
B                                               DK
4510 GOSUB2515:IFB=0THEN4500                    MA
4520 PRINT:GOTO110                              OI
4700 B$="BEGIN SCAN AT ADDRESS":GOSUB2010:AD=
B                                               FH
4705 GOSUB2515:IFB=0THEN4700                    NK
4706 PRINT:GOTO4740                             DI
4710 FORB=0TO7:AC=PEEK(AD+B):GOSUB3030:IFAD+B
=ENTHENAD=SR:GOSUB1080:GOTO110                  BK
4715 PRINT" ";:NEXTB                            EC
4720 PRINT:AD=AD+8                              GN
4730 GETB$:IFB$=CHR$(136)THEN110                MN
4740 GOSUB3010:PRINT": ";:GOTO4710              JD
```

Falling Needles & Shared Birthdays

Bug Repellent codes on program lines are for 64 only! See VIC codes on page 91.

FROM PAGE 53

NEEDLES

```
·1 REM                                   JD
·2 REM  --   NEEDLES   --                EE
·3 REM   RUPERT REPORT #20               OJ
·4 REM                                   JD
·10 REM - SIMULATE THE BUFFON NEEDLE     DC
·11 REM   EXPERIMENT.  DROP NEEDLES OF   GI
·12 REM   LENGTH L ON FLOORBOARDS OF     GP
·13 REM   WIDTH H.  SUCCESS IF NEEDLE    OJ
·14 REM   LANDS ON A CRACK. PROBABILITY  MO
·15 REM   OF LANDING ON A CRACK IS       DC
·16 REM   (2*L)/(H*PI)                   DL
·17 REM                                  JD
·20 H=1 :L=H :TRIES=1 :WINS=1 :K=2*L/H   KF
·30 PRINT CHR$(147)TAB(12)"< TRIES"      NF
·40 PRINT TAB(12)"< PI"                  AJ
·50 PRINT TAB(12)"< AVG"                 DG
·60 Y0=H*RND(0)                          GN
·70 A=RND(0)*30000                       MP
·80 Y=L*SIN(A)+Y0                        IG
·90 IF Y>H OR Y<0 THEN WINS=WINS+1       JG
·100 PI=K*TRIES/WINS                     NC
·110 TTL=TTL+PI : AVG=TTL/TRIES          KE
·120 PRINT CHR$(19)TRIES : PRINT PI      EH
·130 PRINT AVG                           HM
·140 TRIES=TRIES+1                       MN
·150 GOTO 60                             PG
```

BIRTHDAY CALC

```
·1 REM                                   JD
·2 REM  --  BIRTHDAY CALC  --            ML
·3 REM    RUPERT REPORT #20              OJ
·4 REM                                   JD
·10 REM - CALCULATE THE PROBABILITY      CK
·11 REM   THAT FOR A GROUP OF NP PEOPLE  EO
·12 REM   TWO OR MORE OF THEM WERE BORN  MP
·13 REM   ON THE SAME DAY OF THE YEAR    OK
·14 REM                                  JD
·20 PRINT CHR$(147)                      FG
·30 PRINT "HOW MANY PEOPLE ARE IN THE ROO
M":PRINT"(ENTER 0 FOR COMPLETE LIST)"    KI
·40 INPUT NP                             CC
·50 IF NP<=0 OR INT(NP)<>NP THEN 200     BG
·60 PF=1    :REM PROBABILITY OF FAILURE  DM
·70 FOR Q=1 TO NP                        AI
·80 PF=(367-Q)*PF/366 : NEXT             JC
·90 PS=1-PF :REM PROBABILITY OF SUCCESS  IL
·100 PRINT                               JJ
·110 PRINT"IN A GROUP OF" NP "PEOPLE,"   KE
·120 PRINT"THERE IS A" PS*100 "PERCENT"  CG
·130 PRINT"PROBABILITY THAT TWO OR MORE" KG
·140 PRINT"PEOPLE HAVE THE SAME BIRTHDAY" AN
·150 PRINT                               JJ
·160 GOTO 30                             OJ
·200 PF=1                                HM
·210 FOR NP=1 TO 366                     EP
·220 PF=PF*(367-NP)/366                  DA
·230 PRINT "# PEOPLE ="NP;               CO
·240 PRINT "[3" "]% PROB. =" (1-PF)*100  IB
·245 REM  COUNT # OF LINES LISTED (NL)   AC
·250 NL=NL+1 : IF NL<20 THEN 290         GD
·260 NL=0                                IB
·270 PRINT "PRESS ANY KEY FOR MORE[3"."]" LE
·280 GET X$:IF X$="" THEN 280            JH
·290 NEXT                                IA
```

BIRTHDAY SIMULATOR

```
·1 REM                                   JD
·2 REM  -- BIRTHDAY SIMULATOR --         BK
·3 REM      RUPERT REPORT #20            OJ
·4 REM                                   JD
·10 REM - SIMULATE GROUPS OF PEOPLE WITH GN
·11 REM   RANDOMLY CHOSEN BIRTHDAYS.     CL
·12 REM   EACH GROUP CONTAINS 'NUMP'     DH
·13 REM   PEOPLE.  BDAY(X) IS THE        DN
·14 REM   BIRTHDAY (1-366) OF THE XTH    OD
·15 REM   PERSON.  CT(X) IS THE COUNT OF HP
·16 REM   HOW MANY PEOPLE IN THE GROUP   DL
·17 REM   HAVE DAY X AS THEIR BIRTHDAY.  LI
·18 REM   SUCCESS=1 IF TWO PEOPLE IN     OC
·19 REM   ONE GROUP HAVE THE SAME        CO
·20 REM   BIRTHDAY.                      BO
·21 REM                                  JD
·30 NUMP=24  :REM << CHANGE THIS         MJ
·40 DIM BDAY(NUMP),CT(366)               LJ
·50 GROUP=1                              IH
·60 SUCCESS=0                            PH
·65 REM - RESET COUNT OF USED BIRTHDAYS  PP
·70 FOR PERSN=1 TO NUMP                  OG
·80 CT(BDAY(PERSN))=0 : NEXT             IE
·85 REM - CHOOSE EACH PERSON'S BIRTHDAY  ND
·90 FOR PERSN=1 TO NUMP                  OG
·100 DAY=INT(366*RND(0))+1               BK
·110 BDAY(PERSN)=DAY                     JK
·120 CT(DAY)=CT(DAY)+1                   PB
·130 NEXT PERSN                          AA
·135 REM - CHECK FOR DUPLICATES          NK
·140 FOR PERSN=1 TO NUMP                 OG
·150 IF CT(BDAY(PERSN))<2 THEN 180       NP
·160 SUCCESS=1                           PG
·170 PERSN=NUMP                          IE
·180 NEXT PERSN                          AA
·190 IF SUCCESS=1 THEN TTL=TTL+1         LK
·200 PRINT GROUP"GROUPS[4" "]";          AM
·210 PRINT TTL*100/GROUP"% SUCCESS"      OB
·220 GROUP=GROUP+1 : GOTO 60             OK
```

VIC 20 BUG REPELLENT LINE CODES FOR FALLING NEEDLES AND SHARED BIRTHDAYS

NEEDLES

# 1:	JD	# 14:	AE	# 70:	MP
# 2:	GN	# 15:	AJ	# 80:	IG
# 3:	AD	# 16:	BE	# 90:	IC
# 4:	JD	# 17:	JD	# 100:	NC
# 10:	FH	# 20:	IE	# 110:	IJ
# 11:	EG	# 30:	HM	# 120:	OI
# 12:	OK	# 40:	OB	# 130:	HK
# 13:	GG	# 50:	CN	# 140:	MN
		# 60:	GN	# 150:	PD

BIRTHDAY CALC

# 1:	JD	# 40:	CH	# 160:	OM
# 2:	LC	# 50:	LK	# 200:	HM
# 3:	PN	# 60:	KL	# 210:	DJ
# 4:	JD	# 70:	PO	# 220:	DA
# 10:	MI	# 80:	JF	# 230:	OH
# 11:	EE	# 90:	AD	# 240:	FE
# 12:	IP	# 100:	JJ	# 245:	MB
# 13:	IF	# 110:	HH	# 250:	MP
# 14:	JD	# 120:	OH	# 260:	IB
# 20:	FB	# 130:	OE	# 270:	JO
# 30:	BL	# 140:	GD	# 280:	KA
		# 150:	JJ	# 290:	IA

BIRTHDAY SIMULATOR

# 1:	JD	# 19:	CG	# 110:	JK
# 2:	OD	# 20:	BA	# 120:	PB
# 3:	AF	# 21:	JD	# 130:	AO
# 4:	JD	# 30:	HG	# 135:	CG
# 10:	AC	# 40:	AO	# 140:	OG
# 11:	GA	# 50:	IH	# 150:	BA
# 12:	NF	# 60:	PH	# 160:	PG
# 13:	FB	# 65:	GP	# 170:	IE
# 14:	ED	# 70:	OG	# 180:	AO
# 15:	GN	# 80:	IH	# 190:	MJ
# 16:	FF	# 85:	OL	# 200:	FN
# 17:	JA	# 90:	OG	# 210:	NK
# 18:	KH	# 100:	BK	# 220:	OG

Sputter Pop Hiss Chop Flap Crash Slurp Mash

FROM PAGE 18

SIMPLE GUNSHOT (C-64)

```
•1 REM SOUND EFFECTS                              CA
•2 REM SIMPLE GUNSHOT                             JJ
•3 REM (NOTE THAT LINES 100 AND ABOVE ARE
 "SUPPORT"--LINES 20-90 MAKE THE SOUND)           KA
•10 GOSUB 900:GOTO 100                            IO
•20 POKE AD(1),5:POKE SR(1),0:POKE FL(1),
 30:POKE FH(1),30                                 LH
•30 POKE VL,15:PRINT "."                          CJ
•40 POKE GT(1),129                                KC
•50 FOR I=0 TO 59:NEXT                            JC
•60 POKE GT(1),0                                  CF
•90 RETURN                                        IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                       KO
•100 PRINT "PRESS SHIFT TO HEAR SOUND";           DB
•110 IF PEEK(653)=0 THEN 110                      GL
•120 GOSUB 20:GOTO 100                            EI
•898 REM ROUTINE TO SET UP VARIABLES              AJ
•899 REM SET UP SOUND CONTROL ADDRESSES           LB
•900 FL(0)=54272:FH(0)=54273:GT(0)=54276:
AD(0)=54277:SR(0)=54278                           LD
•905 FOR I=1 TO 2:X=I*7:FL(I)=FL(0)+X:FH(
I)=FH(0)+X:GT(I)=GT(0)+X:AD(I)=AD(0)+X            MG
•906 SR(I)=SR(0)+X:NEXT                           BE
•909 REM SET VOLUME ADDRESS                       GJ
•910 VL=54296                                     FN
•990 RETURN                                       IM
```

HELICOPTER BLADES (C-64)

```
•1 REM SOUND EFFECTS                              CA
•2 REM HELICOPTER BLADES                          CD
•3 REM HUMMING SOUND IN VOICE 0 SUGGESTS
 YOU ARE IN HELICOPTER                            LB
•4 REM TO HEAR MORE DISTANT CHOPPER, REMO
 VE ":POKE GT(0),17" FROM LINE 40                 ML
•10 GOSUB 900:GOTO 100                            IO
•20 FOR I=1 TO 2:POKE AD(I),48:POKE SR(I)
,16:NEXT:POKE FH(1),20:POKE FH(2),40              BC
•25 POKE AD(0),34:POKE SR(0),49:POKE FH(0
),10                                              LD
•30 POKE VL,15:PRINT "PRESS SHIFT TO STOP
 SOUND"                                           OC
•40 POKE GT(1),129:POKE GT(2),129:POKE GT
(0),17                                            KF
•50 FOR I=10 TO 220 STEP 70:POKE FH(1),I:
POKE FH(2),I:NEXT                                 NP
•60 POKE GT(1),0:POKE GT(2),0:POKE GT(0),
0:IF PEEK(653)=0 THEN 40                          LM
•70 IF PEEK(653)<>0 THEN 70                       AK
•90 RETURN                                        IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                       KO
•100 PRINT "PRESS SHIFT TO HEAR SOUND"            OL
•110 IF PEEK(653)=0 THEN 110                      GL
•120 GOSUB 20:GOTO 100                            EI
•898 REM ROUTINE TO SET UP VARIABLES              AJ
•899 REM SET UP SOUND CONTROL ADDRESSES           LB
•900 FL(0)=54272:FH(0)=54273:GT(0)=54276:
AD(0)=54277:SR(0)=54278                           LD
•905 FOR I=1 TO 2:X=I*7:FL(I)=FL(0)+X:FH(
I)=FH(0)+X:GT(I)=GT(0)+X:AD(I)=AD(0)+X            MG
•906 SR(I)=SR(0)+X:NEXT                           BE
•909 REM SET VOLUME ADDRESS                       GJ
•910 VL=54296                                     FN
•990 RETURN                                       IM
```

SAWING WOOD (C-64)

```
•1 REM SOUND EFFECTS                              CA
•2 REM SAWING WOOD                                LE
•10 GOSUB 900:GOTO 100                            IO
•20 FOR I=1 TO 2:POKE AD(I),0 :POKE SR(I)
,243:NEXT                                         JN
•25 POKE AD(0),64:POKE SR(0),66                   IH
•30 POKE FH(1),255:POKE FH(2),12 :POKE FH
(0),200                                           GD
•35 POKE VL,15:PRINT "."                          CJ
•40 POKE GT(1),129:POKE GT(2),129:POKE GT
```

```
(0),21                                          KK
•45 FOR I=0 TO 7:POKE VL,5:POKE VL,15:FOR
 X=0 TO INT(4*RND(9)):NEXT:NEXT                 AI
•50 FOR I=15 TO 0 STEP-1:POKE VL,I:NEXT         BH
•55 FOR I=1 TO 15:POKE VL,I:FOR X=0 TO 3:
NEXT:NEXT                                       AK
•60 FOR I=0 TO 2:POKE GT(I),0:NEXT              KA
•90 RETURN                                      IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                     KO
•100 PRINT "PRESS SHIFT TO HEAR SOUND";         DB
•110 IF PEEK(653)=0 THEN 110                    GL
•120 GOSUB 20:GOTO 100                          EI
•898 REM ROUTINE TO SET UP VARIABLES            AJ
•899 REM SET UP SOUND CONTROL ADDRESSES         LB
•900 FL(0)=54272:FH(0)=54273:GT(0)=54276:
AD(0)=54277:SR(0)=54278                         LD
•905 FOR I=1 TO 2:X=I*7:FL(I)=FL(0)+X:FH(
I)=FH(0)+X:GT(I)=GT(0)+X:AD(I)=AD(0)+X          MG
•906 SR(I)=SR(0)+X:NEXT                         BE
•909 REM SET VOLUME ADDRESS                     GJ
•910 VL=54296                                   FN
•990 RETURN                                     IM
```

SPACE GUNSHOT (C-64)

```
•1 REM SOUND EFFECTS                            CA
•2 REM "SPACE" GUNSHOT                          LE
•10 GOSUB 900:GOTO 100                          IO
•20 POKE AD(1),5:POKE SR(1),53:POKE FL(1)
,30:POKE FH(1),30                               PJ
•25 POKE AD(0),5:POKE SR(0),53 :POKE FH(0
),250                                           NG
•30 POKE VL,15:PRINT "."                        CJ
•40 POKE GT(1),129:FOR I=0 TO 29:NEXT:POK
E GT(0),17                                      AG
•50 FOR I=225 TO 25 STEP -25:POKE FH(0),I
:FOR X=0 TO 19:NEXT:NEXT                        JJ
•60 POKE GT(1),0:POKE GT(0),0                   ID
•90 RETURN                                      IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                     KO
•100 PRINT "PRESS SHIFT TO HEAR SOUND";         DB
•110 IF PEEK(653)=0 THEN 110                    GL
•120 GOSUB 20:GOTO 100                          EI
•898 REM ROUTINE TO SET UP VARIABLES            AJ
•899 REM SET UP SOUND CONTROL ADDRESSES         LB
•900 FL(0)=54272:FH(0)=54273:GT(0)=54276:
AD(0)=54277:SR(0)=54278                         LD
•905 FOR I=1 TO 2:X=I*7:FL(I)=FL(0)+X:FH(
I)=FH(0)+X:GT(I)=GT(0)+X:AD(I)=AD(0)+X          MG
•906 SR(I)=SR(0)+X:NEXT                         BE
•909 REM SET VOLUME ADDRESS                     GJ
•910 VL=54296                                   FN
•990 RETURN                                     IM
```

SEASHORE (C-64)

```
•1 REM SOUND EFFECTS                            CA
•2 REM SEASHORE                                 PL
•10 GOSUB 900:GOTO 100                          IO
•20 POKE AD(1),160:POKE SR(1),255:POKE FL
(1),30:POKE FH(1),30                            AF
•25 POKE AD(2),15:POKE SR(2),0:POKE FH(2)
,80                                             LH
•30 POKE VL,15:PRINT "PRESS RUNSTOP/RESTO
RE TO STOP"                                     GC
•35 POKE GT(1),129                              KC
•40 POKE GT(2),129:FOR I=255 TO 100 STEP
-(1+INT(3*RND(9))):POKE FH(1),I                 LE
•50 X=4+INT(I/23):IF X>15 THEN X=15             DA
•55 POKE VL,X:FOR X=0 TO 29:NEXT:NEXT:POK
E GT(2),0                                       NI
•60 FOR I=100 TO 255 STEP 8+INT(10*RND(9)
):POKE FH(1),I                                  FD
•70 X=4+INT(I/23):IF X>15 THEN X=15             DA
•75 POKE VL,X:FOR X=0 TO 19:NEXT:NEXT           EA
•80 GOTO 40                                     PE
•90 RETURN                                      IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                     KO
•100 PRINT "PRESS SHIFT TO HEAR SOUND"          OL
•110 IF PEEK(653)=0 THEN 110                    GL
•120 GOSUB 20:GOTO 100                          EI
•898 REM ROUTINE TO SET UP VARIABLES            AJ
•899 REM SET UP SOUND CONTROL ADDRESSES         LB
•900 FL(0)=54272:FH(0)=54273:GT(0)=54276:
AD(0)=54277:SR(0)=54278                         LD
•905 FOR I=1 TO 2:X=I*7:FL(I)=FL(0)+X:FH(
I)=FH(0)+X:GT(I)=GT(0)+X:AD(I)=AD(0)+X          MG
•906 SR(I)=SR(0)+X:NEXT                         BE
•909 REM SET VOLUME ADDRESS                     GJ
•910 VL=54296                                   FN
•990 RETURN                                     IM
```

BOINK (C-64)

```
•1 REM SOUND EFFECTS                            CA
•2 REM "BOINK"                                  FF
•10 GOSUB 900:GOTO 100                          IO
•20 POKE AD(1),5:POKE SR(1),21:POKE FL(1)
,30:POKE FH(1),30                               DI
•25 POKE AD(0),5:POKE SR(0),53 :POKE FH(0
),25                                            LJ
•30 POKE VL,15:PRINT "."                        CJ
•40 POKE GT(1),129:FOR I=0 TO 29:NEXT:POK
E GT(0),17                                      AG
•50 FOR I=25 TO 250 STEP 25:POKE FH(0),I:
FOR X=0 TO 14:NEXT:NEXT                         AD
•60 POKE GT(1),0:POKE GT(0),0                   ID
•90 RETURN                                      IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                     KO
•100 PRINT "PRESS SHIFT TO HEAR SOUND";         DB
•110 IF PEEK(653)=0 THEN 110                    GL
•120 GOSUB 20:GOTO 100                          EI
•898 REM ROUTINE TO SET UP VARIABLES            AJ
•899 REM SET UP SOUND CONTROL ADDRESSES         LB
•900 FL(0)=54272:FH(0)=54273:GT(0)=54276:
```

```
 AD(0)=54277:SR(0)=54278                      LD
•905 FOR I=1 TO 2:X=I*7:FL(I)=FL(0)+X:FH(
 I)=FH(0)+X:GT(I)=GT(0)+X:AD(I)=AD(0)+X       MG
•906 SR(I)=SR(0)+X:NEXT                        BE
•909 REM SET VOLUME ADDRESS                    GJ
•910 VL=54296                                  FN
•990 RETURN                                    IM
```

COMPLEX GUNSHOT (C-64)

```
•1 REM SOUND EFFECTS                           CA
•2 REM COMPLEX GUNSHOT                         NA
•3 REM (NOTE THAT LINES 100 AND ABOVE ARE
 "SUPPORT"--LINES 20-90 MAKE THE SOUND)        KA
•10 GOSUB 900:GOTO 100                         IO
•20 POKE AD(1),5:POKE SR(1),53:POKE FL(1)
 ,30:POKE FH(1),30                             PJ
•25 POKE AD(0),5:POKE SR(0),53 :POKE FH(0
 ),250                                         NG
•30 POKE VL,15:PRINT "."                       CJ
•40 POKE GT(1),129:FOR I=0 TO 29:NEXT:POK
 E GT(0),17                                    AG
•50 FOR I=225 TO 25 STEP -25:POKE FH(0),I
 :NEXT                                         LF
•60 POKE GT(1),0:POKE GT(0),0                  ID
•90 RETURN                                     IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                    KO
•100 PRINT "PRESS SHIFT TO HEAR SOUND";        DB
•110 IF PEEK(653)=0 THEN 110                   GL
•120 GOSUB 20:GOTO 100                         EI
•898 REM ROUTINE TO SET UP VARIABLES           AJ
•899 REM SET UP SOUND CONTROL ADDRESSES        LB
•900 FL(0)=54272:FH(0)=54273:GT(0)=54276:
 AD(0)=54277:SR(0)=54278                       LD
•905 FOR I=1 TO 2:X=I*7:FL(I)=FL(0)+X:FH(
 I)=FH(0)+X:GT(I)=GT(0)+X:AD(I)=AD(0)+X        MG
•906 SR(I)=SR(0)+X:NEXT                        BE
•909 REM SET VOLUME ADDRESS                    GJ
•910 VL=54296                                  FN
•990 RETURN                                    IM
```

GUNSHOT (VIC 20)

```
•1 REM SOUND EFFECTS                           OF
•2 REM GUNSHOT (VIC VERSION)                   GJ
•10 GOSUB 900:GOTO 100                         JL
•20 POKE FR(0),0:POKE FR(1),0:POKE FR(2),
 0:POKE FR(3),115                              OB
•30 POKE VL,15:PRINT "."                       AM
•45 FOR I=15 TO 0 STEP -1:POKE VL,I:POKE
 FR(3),P(70+3*I):NEXT                          NO
•60 FOR I=0 TO 2:POKE FR(I),0:NEXT             LJ
•90 RETURN                                     IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                    FN
•100 PRINT "PRESS SHIFT";                      PD
•110 IF PEEK(653)=0 THEN 110                   GF
•120 GOSUB 20:GOTO 100                         EK
•898 REM ROUTINE TO SET UP VARIABLES           JF
•899 REM SET UP SOUND CONTROL ADDRESSES        CF
•900 DIM FR(3),P(127)                          BI
•910 FR(0)=36874:FOR I=1 TO 3:FR(I)=FR(I-
 1)+1:NEXT                                     BC
•920 FOR I=0 TO 127:P(I)=I+128:NEXT            IC
•929 REM SET VOLUME ADDRESS                    GI
•930 VL=36878                                  GP
•990 RETURN                                    IM
```

HELICOPTER (VIC 20)

```
•1 REM SOUND EFFECTS                           OF
•2 REM HELICOPTER (VIC VERSION)                GH
•10 GOSUB 900:GOTO 100                         JL
•20 POKE FR(0),0:POKE FR(1),0:POKE FR(2),
 0:POKE FR(3),P(50)                            EP
•30 POKE VL,0                                  KE
•40 FOR I=0 TO 15:POKE VL,I:NEXT               DG
•45 FOR I=0 TO 2:NEXT                          FI
•50 POKE FR(3),60:FOR I=15 TO 0 STEP-1:PO
 KE VL,I:NEXT                                  GB
•60 FOR I=0 TO 3:POKE FR(I),0:NEXT             NE
•90 RETURN                                     IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                    FN
•100 PRINT "PRESS SHIFT TO STOP"               KK
•110 PRINT".";:IF PEEK(653)>0 THEN END         LI
•120 GOSUB 20:GOTO 110                         EP
•898 REM ROUTINE TO SET UP VARIABLES           JF
•899 REM SET UP SOUND CONTROL ADDRESSES        CF
•900 DIM FR(3),P(127)                          BI
•910 FR(0)=36874:FOR I=1 TO 3:FR(I)=FR(I-
 1)+1:NEXT                                     BC
•920 FOR I=0 TO 127:P(I)=I+128:NEXT            IC
•929 REM SET VOLUME ADDRESS                    GI
•930 VL=36878                                  GP
•990 RETURN                                    IM
```

SPACE GUNSHOT (VIC 20)

```
•1 REM SOUND EFFECTS                           OF
•2 REM SPACE GUNSHOT (VIC VERSION)             MD
•10 GOSUB 900:GOTO 100                         JL
•20 POKE FR(0),0:POKE FR(1),0:POKE FR(2),
 0:POKE FR(3),115                              OB
•30 PRINT "."                                  AI
•45 FOR I=15 TO 0 STEP -1.5:POKE VL,I:X=P
 (70+3*I):POKE FR(3),X                         OF
•50 POKE FR(0),450-X:NEXT:POKE FR(3),0         PH
•55 FOR I=1 TO 9 STEP 3:POKE VL,I:NEXT         BK
•60 FOR I=0 TO 3:POKE FR(I),0:NEXT             NE
•90 RETURN                                     IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
 IS PRESSED                                    FN
•100 PRINT "PRESS SHIFT";                      PD
•110 IF PEEK(653)=0 THEN 110                   GF
•120 GOSUB 20:GOTO 100                         EK
•898 REM ROUTINE TO SET UP VARIABLES           JF
•899 REM SET UP SOUND CONTROL ADDRESSES        CF
•900 DIM FR(3),P(127)                          BI
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 87 and 88 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
•910 FR(0)=36874:FOR I=1 TO 3:FR(I)=FR(I-
 1)+1:NEXT                                BC
•920 FOR I=0 TO 127:P(I)=I+128:NEXT       IC
•929 REM SET VOLUME ADDRESS               GI
•930 VL=36878                             GP
•990 RETURN                               IM
```

BOINK (VIC 20)

```
•1 REM SOUND EFFECTS                      OF
•2 REM BOINK (VIC VERSION)                KH
•10 GOSUB 900:GOTO 100                    JL
•20 POKE FR(0),P(20):POKE FR(1),0:POKE FR
 (2),0:POKE FR(3),P(20)                   EL
•30 PRINT "."                             AI
•40 FOR I=15 TO 0 STEP -1:POKE VL,I:NEXT:
 POKE FR(3),0                             CO
•50 FOR I=1 TO 15 STEP 2:POKE FR(0),P(100
 +I):POKE VL,I:NEXT                       CI
•60 FOR I=0 TO 3:POKE FR(I),0:NEXT:POKE V
 L,0                                      HL
•90 RETURN                                IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
  IS PRESSED                              FN
•100 PRINT "PRESS SHIFT";                 PD
•110 IF PEEK(653)=0 THEN 110              GF
•120 GOSUB 20:GOTO 100                    EK
•898 REM ROUTINE TO SET UP VARIABLES      JF
•899 REM SET UP SOUND CONTROL ADDRESSES   CF
•900 DIM FR(3),P(127)                     BI
•910 FR(0)=36874:FOR I=1 TO 3:FR(I)=FR(I-
 1)+1:NEXT                                BC
•920 FOR I=0 TO 127:P(I)=I+128:NEXT       IC
•929 REM SET VOLUME ADDRESS               GI
•930 VL=36878                             GP
•990 RETURN                               IM
```

SAWING WOOD (VIC 20)

```
•1 REM SOUND EFFECTS                      OF
•2 REM SAWING WOOD (VIC VERSION)          ON
•10 GOSUB 900:GOTO 100                    JL
•20 POKE FR(0),P(0):POKE FR(1),0:POKE FR(
 2),0:POKE FR(3),20                       ND
•30 POKE VL,15:PRINT "."                  AM
•45 FOR I=0 TO 5:POKE VL,5:POKE VL,15:FOR
  X=0 TO INT(3*RND(9)):NEXT:NEXT          FD
•50 FOR I=15 TO 0 STEP-1:POKE VL,I:NEXT   KD
•55 FOR I=1 TO 15:POKE VL,I:FOR X=0 TO 2:
 NEXT:NEXT                                FP
•60 FOR I=0 TO 2:POKE FR(I),0:NEXT        LJ
•90 RETURN                                IM
•99 REM ROUTINE TO PLAY SOUNDS WHEN SHIFT
  IS PRESSED                              FN
•100 PRINT "PRESS SHIFT";                 PD
•110 IF PEEK(653)=0 THEN 110              GF
•120 GOSUB 20:GOTO 100                    EK
•898 REM ROUTINE TO SET UP VARIABLES      JF
•899 REM SET UP SOUND CONTROL ADDRESSES   CF
•900 DIM FR(3),P(127)                     BI
•910 FR(0)=36874:FOR I=1 TO 3:FR(I)=FR(I-
 1)+1:NEXT                                BC
•920 FOR I=0 TO 127:P(I)=I+128:NEXT       IC
•929 REM SET VOLUME ADDRESS               GI
•930 VL=36878                             GP
•990 RETURN                               IM
```

SOUND-A-RAMA

FROM PAGE 27

```
•30 PRINT"[CLEAR]"SPC(14)"[RVSON]SOUND-A-
 RAMA[RVSOFF]"                            AB
•40 PRINT"[DOWN]IF YOU'RE USING THE DOS W
 EDGE, PLEASE[3" "]DISABLE IT BY ENTERING
 "                                        MK
•50 PRINTSPC(18)"[DOWN]@Q"                ME
•60 PRINT"[DOWN]THIS WILL MAKE THE PROGRA
 M RUN FASTER."                           AA
•70 PRINT"[DOWN]THEN ENTER":PRINTSPC(15)"
 [DOWN]RUN 110"                           IJ
•80 END                                   IC
•100 REM SOUND-A-RAMA BY BRUCE BARTLETT
      OCT. 1984                           LP
•110 GOTO 1520                            FN
•120 REM ***COMPUTER TAPE RECORDER--MAIN
 LOOP***                                  OF
•130 GETC$(N):IFC$(N)=""THEN 130          FG
•140 IFC$(N)="P"THENIFN>UNTHENPRINT"[CLEA
 R][DOWN][WHITE]PLAYING[3"."]":M=N:POKES+
 4,16:N=1:GOTO 180                        DJ
•150 IFC$(N)="M"THENRUN110                LC
•160 POKES+F,XT:D(N-UN)=TI:TI$="[6"0"]":P
 OKES+F,SE:POKES+UN,TN*VAL(C$(N))         DC
•170 N=N+UN:GOTO 130                      PL
•180 POKES+F,SE:POKES+UN,TN*VAL(C$(N)):FO
 RT=UNTOTT*D(N):NEXT:POKES+F,XT           IE
•190 N=N+UN:IFN=MTHENRUN1860              JJ
•200 GOTO 180                             CN
•210 REM  ***DRUM MACHINE--MAIN LOOP***   HO
•220 GETA$                                AL
•230 IFA$="M"THENPOKE650,0:RUN110         HB
•240 ONVAL(A$)GOTO 270, 300, 330, 360, 39
 0, 420                                   EI
•250 GOTO 220                             BM
•260 REM ***HI HAT***                     CC
•270 POKEFU,TF:POKEFL,TF:POKEAD,FV:POKESR
 ,FV                                      KE
•280 POKEW,WN:FORT=UNTOFY:NEXT:POKEW,WE:G
 OTO 220                                  FK
•290 REM ***SNARE***                      PH
•300 POKEFU,SY:POKEFL,.:POKEAD,.:POKESR,T
 V                                        JP
•310 POKEW,WN:FORT=UNTOFY:NEXT:POKEW,WE:G
 OTO 220                                  FK
```

```
·320 REM ***HI TOM***                              BD
·330 POKEFU,ET:POKEFL,.:POKEAD,.:POKESR,T
 N:POKEV,TL:POKEV,FT                               AC
·340 POKEW,SE:FORT=UNTOFY:NEXT:POKEW,XT:G
 OTO 220                                           PH
·350 REM ***LOW TOM***                             GP
·360 POKEFU,SX:POKEFL,.:POKEAD,.:POKESR,T
 N:POKEV,TL:POKEV,FT                               KA
·370 POKEW,SE:FORT=UNTOFY:NEXT:POKEW,XT:G
 OTO 220                                           PH
·380 REM ***KICK DRUM***                           MC
·390 POKEFU,TE:POKEFL,. :POKEAD,.:POKESR,
 TR:POKEV,TH:POKEV,FT                              JM
·400 POKEW,SE:FORT=UNTOFY:NEXT:POKEW,XT:G
 OTO 220                                           PH
·410 REM ***CYMBAL***                              GD
·420 POKEFU(2),TF:POKEFL(2),TF:POKEAD(2),
 NN:POKESR(2),NN                                   JI
·430 POKEW(2),WN:FORT=UNTOFY:NEXT:POKEW(2
 ),WE:GOTO 220                                     GP
·440 REM ***WINDING DOWN RECORD***                 HB
·450 D=60                                          GB
·460 FORN=1TO1.75STEP.25:POKEW,33:POKEW,1
 7:POKEFU,30*N-.04*D:FORT=1TOD:NEXT                JF
·470 D=D*1.05                                      KH
·480 IFD>700THENFORF=16TO0STEP-.08:POKEFU
 ,F:NEXT:POKEW,16:GOTO 1760                        GL
·490 NEXTN                                         NA
·500 POKEW,16:GOTO 460                             AK
·510 REM ***PULVERIZER***                          ND
·520 POKEW,19                                      II
·530 POKESU,32                                     OA
·540 FORN=1TO20                                    MH
·550 FORF=255TO1STEP-N:POKEFU,F:NEXTF              EL
·560 NEXTN                                         NA
·570 POKEW,18                                      IJ
·580 GOTO1760                                      GD
·590 REM ***DISINTEGRATOR***                       HD
·600 POKESL,67:POKESU,12                           MN
·610 POKEW,21                                      IH
·620 FORR=1TO3:FORN=1TO40                          JB
·630 FORF=0TO255STEPN+5:POKEFU,F:NEXTF             KL
·640 NEXTN,R                                       FH
·650 POKEW,20:GOTO 1760                            BE
·660 REM ***OLD WASHING MACHINE***                 HF
·670 FORN=1TO20                                    MH
·680 FORF=1TO64                                    MP
·690 POKEW,17:POKEFU,F:FORT=1TO15:NEXT:PO
 KEW,32:FORT=1TO5:NEXT                             EL
·700 F=F*2:NEXTF,N:GOTO 1760                       DC
·710 REM ***ENERGY LEVELS***                       AJ
·720 POKEAD,9:POKESR,64+12                         NP
·730 POKESU,3:POKESL,220                           NK
·740 POKEFL,10                                     LK
·750 POKEW,21                                      IH
·760 FORT=1TO2500:POKEFU,T/100:NEXT                GJ
·770 POKEW,20:GOTO 1760                            BE
·780 REM ***NERVOUS***                             PD
·790 POKEAD,63:POKEFU,4:POKEFL,10:POKESU,
 10                                                NB
·800 POKEW,21                                      IH
·810 FORN=1TO25                                    MC
·820 FORF=0TO255STEP20:POKESL,F:NEXTF              DJ
·830 FORF=255TO0STEP-20:POKESL,F:NEXTF             JH
·840 NEXTN                                         NA
·850 POKEW,20:GOTO 1760                            BE
·860 REM ***SPROING***                             MH
·870 POKESR,15*16+9                                NB
·880 POKEW,33:POKEFU,3:FORT=1TO150:NEXT            BJ
·890 POKEFU,67:POKEW,19                            IK
·900 FORN=1TO50                                    MM
·910 FORF=0TO50STEPN:POKESU,F:NEXTF,N              DN
·920 POKEW,18:FORT=1TO500:NEXT:GOTO 1760           NB
·930 REM ***FORCE FIELD***                         CM
·940 POKEAD,13*16+15:POKESR,15*16+15               GC
·950 POKESU,3:POKESL,220                           NK
·960 POKEFU,23:POKEFL,10:POKEW,21                  IF
·970 FORT=1TO5000:NEXT:FORL=15TO1STEP-.05
 :POKEV,L:NEXT                                     LI
·980 POKEW,0:FORT=1TO500:NEXT:POKEV,15:GO
 TO 1760                                           OC
·990 REM ***INVOLUTION***                          MG
·1000 POKEV,15+32:POKES+23,239                     PF
·1010 POKEW,33                                     IE
·1020 FORSP=1OTO255                                GH
·1030 FORF=255TO0STEP-SP:POKEFU,F:POKES+2
 2,F:NEXT                                          IH
·1040 NEXTSP                                       CC
·1050 POKEW,32:POKES+23,0:GOTO 1760                JK
·1060 REM ***AVIARY***                             HD
·1070 FORLOOP=1TO2:POKEAD,119:FORN=1TO15:
 POKEW,17                                          KA
·1080 FORF=INT(RND(.)*50)+50TOINT(RND(.)*
 150)+50STEP2:POKEFU,F:NEXTF:POKEW,16              JL
·1090 NEXTN                                        NA
·1100 POKEV,12:POKEAD,15                           EJ
·1110 POKEW,17:FORN=1TOINT(RND(1)*10)+8:F
 ORF=50TO200STEP20:POKEFU,F                        JJ
·1120 NEXTF,N                                      EL
·1130 FORN=1TOINT(RND(1)*10)+8:FORF=50TO2
 00STEP10:POKEFU,F                                 IB
·1140 NEXTF,N                                      EL
·1150 POKEW,16:POKEV,15:NEXTLOOP:GOTO 176
 0                                                 KE
·1160 REM ***GHOST***                              BP
·1170 FORN=1TO2                                    JA
·1180 FORFS=100TO0STEP-5:POKEW,17                  NC
·1190 FORF=30TO5STEP-2:POKEFU,F+FS:NEXT            BA
·1200 FORF=5TO30STEP2:POKEFU,F+FS:NEXT             LK
·1210 POKEW,16:NEXTFS,N                            FB
·1220 POKEW,16:GOTO 1760                           DB
·1230 REM ***SONG OF SATURN***                     CN
·1240 POKEAD,16*11:POKESR,16*15+11:POKES+
 23,239:POKEV,15+32                                MP
·1250 FORN=1TO10:POKEFU,INT(RND(1)*10)+1           AF
·1260 FORSW=0TOINT(RND(1)*200):POKEW,129:
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 87 and 88 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
POKES+22,SW:NEXTSW                              HK
·1270 POKEW,128:FORT=1TO400:NEXT                KL
·1280 NEXTN                                     NA
·1290 FORT=1TO1500:NEXT:POKES+23,0:POKEV,
15:GOTO 1760                                    EO
·1300 REM ***KETCHUP SQUIRTS***                 JC
·1310 POKEAD,15:POKES+23,239:POKEV,15+32        LJ
·1320 FORN=1TO10                                MI
·1330 POKEW,33                                  IE
·1340 FORSW=0TO150STEPINT(RND(1)*20)+3          PE
·1350 POKES+22,SW:POKEFU,INT(RND(1)*10):N
EXT                                             DG
·1360 POKEW,32:FORT=1TO400:NEXT                 GC
·1370 NEXTN                                     NA
·1380 POKES+23,0:POKEV,15:GOTO 1760             IG
·1390 REM ***ELF LAUGH***                       LC
·1400 FORL=1TO3:F=L*10+50                       EI
·1410 FORN=1TO10:POKEW,33                       FJ
·1420 POKEFU,F:FORT=1TO3:NEXT:POKEW,128:F
ORT=1TO60:NEXT                                  KP
·1430 F=F-5                                     BM
·1440 NEXTN,L                                   FB
·1450 GOTO 1760                                 GD
·1460 REM ***THROB***                           AP
·1470 POKEFU,1                                  JN
·1480 FORF=255TO180STEP-.06                     EN
·1490 POKEFL,F:POKEW,33:POKEW,17:NEXT           JA
·1500 POKEW,32:GOTO 1760                        DH
·1510 REM ***INITIALIZATION***                  IJ
·1520 PRINT"[CLEAR]":POKE53281,0:POKE5328
0,10                                            OH
·1530 S=54272:FORL=0TO24:POKES+L,.:NEXT         OD
·1540 V=S+24:W=S+4:FU=S+1:FL=S:AD=S+5:SR=
S+6:SL=S+14:SU=S+15:POKEV,15                    ME
·1550 W(2)=S+11:FU(2)=S+8:FL(2)=S+7:AD(2)
=S+12:SR(2)=S+13                                BL
·1560 PRINTSPC(10)"[DOWN][GREEN][3"*"][RV
SON][YELLOW]SOUND-A-RAMA[RVSOFF][GREEN][
3"*"][c 7]"                                     JO
·1570 PRINT:PRINT                               GJ
·1580 PRINT"[RVSON]A[RVSOFF] COMPUTER TAP
E RECORDER                                      EE
·1590 PRINT"[RVSON]B[RVSOFF] DRUM MACHINE
"                                               PH
·1600 PRINT"[RVSON]C[RVSOFF] WINDING DOWN
 RECORD"                                        MF
·1610 PRINT"[RVSON]D[RVSOFF] PULVERIZER"        OJ
·1620 PRINT"[RVSON]E[RVSOFF] DISINTEGRATO
R"                                              OF
·1630 PRINT"[RVSON]F[RVSOFF] OLD WASHING
MACHINE"                                        MA
·1640 PRINT"[RVSON]G[RVSOFF] ENERGY LEVEL
S"                                              CK
·1650 PRINT"[RVSON]H[RVSOFF] NERVOUS"           NE
·1660 PRINT"[RVSON]I[RVSOFF] SPROING"           OJ
·1670 PRINT"[RVSON]J[RVSOFF] FORCE FIELD"       KK
·1680 PRINT"[RVSON]K[RVSOFF] INVOLUTION"        CD
·1690 PRINT"[RVSON]L[RVSOFF] AVIARY"            LD
·1700 PRINT"[RVSON]M[RVSOFF] GHOST"             GN
·1710 PRINT"[RVSON]N[RVSOFF] SONG OF SATU
RN"                                             LF
·1720 PRINT"[RVSON]O[RVSOFF] KETCHUP SQUI
RTS"                                            EF
·1730 PRINT"[RVSON]P[RVSOFF] ELF LAUGH"         OE
·1740 PRINT"[RVSON]Q[RVSOFF] THROB"             EN
·1750 PRINT"[DOWN][CYAN][5" "]TYPE THE LE
TTER OF YOUR CHOICE[c 7]"                       AL
·1760 POKEFU,0:POKEFL,0:POKEAD,15:POKESR,
0                                               CK
·1770 IFA>0ANDA<18THENPRINT"[HOME]":FORN=
1TOA+3:PRINT:NEXT:PRINT"[RVSON][c 7]"CHR
$(A+64)"[RVSOFF]"                               DA
·1780 GETA$:IFA$=""THEN1780                     LO
·1790 A=ASC(A$)-64:IFA<1ORA>17THEN 1780         PA
·1800 PRINT"[HOME]":FORN=1TOA+3:PRINT:NEX
T:PRINT"[RVSON][YELLOW]"CHR$(A+64)"[RVSO
FF]"                                            DF
·1810 IFA=16THEN 1400                           IF
·1820 IFA=17THEN 1470                           KH
·1830 ONAGOTO1860,1920,450,520,600,670,72
0,790,870,940,1000,1070,1170,1240,1310          KC
·1840 GOTO 1780                                 HI
·1850 REM ***COMPUTER TAPE RECORDER INITI
ALIZATION***                                    HC
·1860 DIMC$(500),D(500):SE=17:XT=16:UN=1:
F=4:TN=10:TT=13:N=1:S=54272:POKES+5,9           MC
·1870 POKE53281,9:POKE53280,4:PRINT"[CLEA
R]"SPC(8)"[RVSON][YELLOW]COMPUTER TAPE R
ECORDER[RVSOFF]"                                MH
·1880 PRINT"[DOWN][DOWN][CYAN]PLAY MUSIC
ON KEYS 1-9.":PRINT"[DOWN]HIT [RVSON]SPA
CE[RVSOFF] WHEN DONE."                          FI
·1890 PRINT"[DOWN]HIT 'P' TO PLAY THE REC
ORDING."                                        DL
·1900 PRINT"[DOWN]HIT 'M' TO RETURN TO MA
IN MENU.":GOTO 130                              JM
·1910 REM ***DRUM MACHINE INITIALIZATION*
**                                              NM
·1920 TF=255:UN=1:TW=2:NN=9:FV=5:FY=50:SE
=17:XT=16:WN=129:WE=128                         DO
·1930 SY=60:TV=245:ET=8:TL=12:FT=15:SX=6:
TN=249:TR=244:TH=13:TE=3:POKE650,128            DH
·1940 POKE53281,4:POKE53280,6:PRINT"[CLEA
R]"SPC(13)"[RVSON][YELLOW]DRUM MACHINE[R
VSOFF]"                                         AG
·1950 PRINT"[DOWN]1=HI HAT":PRINT"[DOWN]2
=SNARE":PRINT"[DOWN]3=HI TOM":PRINT"[DOW
N]4=LOW TOM"                                    FP
·1960 PRINT"[DOWN]5=KICK DRUM":PRINT"[DOW
N]6=CYMBAL":PRINT"[DOWN][DOWN]M = MAIN M
ENU"                                            KJ
·1970 GOTO 220                                  EE
```

WINDOWS

FROM PAGE 36 C-64 VERSION

```
·110 PRINT"[CLEAR]":POKE53280,0:POKE53281
 ,7                                      OI
·120 PRINT"[UP][RVSON][c 2][12" "][RVSOFF
 ][sEP][14" "][c *][RVSON][12" "]"       AE
·130 PRINT"[UP][RVSON][11" "][RVSOFF][sEP
 ][16" "][c *][RVSON][11" "]"            HP
·140 PRINT"[UP][RVSON][10" "][RVSOFF][sEP
 ][18" "][c *][RVSON][10" "]"            FF
·150 PRINT"[UP][RVSON][9" "][RVSOFF][sEP]
 [20" "][c *][RVSON][9" "]"              DN
·160 PRINT"[UP][RVSON][8" "][RVSOFF][sEP]
 [22" "][c *][RVSON][8" "]"              FD
·170 PRINT"[UP][RVSON][7" "][RVSOFF][sEP]
 [24" "][c *][RVSON][7" "]"              LP
·180 PRINT"[UP][RVSON][6" "][RVSOFF][sEP]
 [26" "][c *][RVSON][6" "]"              FF
·190 PRINT"[UP][RVSON][5" "][RVSOFF][sEP]
 [28" "][c *][RVSON][5" "]"              DN
·200 PRINT"[UP][RVSON][4" "][RVSOFF][sEP]
 [BLACK][3" "][c N]  [c H][s -][c N][s M]
  [c H][s O][s M] [s O][s P][c N]  [c H][
 c O] [s L] [s N][c H][3" "][c 2][c *][RV
 SON][4" "]"                             DJ
·210 PRINT"[UP][RVSON][BLACK][4" "][RVSOF
 F][BLACK][4" "][c L][s N][s M][c H][s -]
 [c L] [s M][c H][s L][s N][c L][s L][s @
 ][c L][s N][s M][c H][c U][c N][s L][c H
 ][s P][c G][4" "][RVSON][4" "]"         HE
·220 PRINT"[UP][RVSON][c 2][4" "][c *][RV
 SOFF][30" "][RVSON][sEP][4" "]"         BM
·230 PRINT"[UP][RVSON][5" "][c *][RVSOFF]
 [28" "][RVSON][sEP][5" "]"              BF
·240 PRINT"[UP][RVSON][6" "][c *][RVSOFF]
 [26" "][RVSON][sEP][6" "]"              FH
·250 PRINT"[UP][RVSON][7" "][c *][RVSOFF]
 [24" "][RVSON][sEP][7" "]"              AP
·260 PRINT"[UP][RVSON][8" "][c *][RVSOFF]
 [22" "][RVSON][sEP][8" "]"              FJ
·270 PRINT"[UP][RVSON][9" "][c *][RVSOFF]
 [20" "][RVSON][sEP][9" "]"              AN
·280 PRINT"[UP][RVSON][10" "][c *][RVSOFF
 ][18" "][RVSON][sEP][10" "]"            FH
·290 PRINT"[UP][RVSON][11" "][c *][RVSOFF
 ][16" "][RVSON][sEP][11" "]"            HP
·300 PRINT"[UP][RVSON][12" "][c *][RVSOFF
 ][14" "][RVSON][sEP][12" "]"            FJ
·310 PRINT"[UP][BLUE][40"[c +]"]"        IA
·320 PRINTTAB(13)"BY THOMAS BUNKER"      HO
·330 PRINTTAB(12)"[DOWN]AND THE AHOY! STA
 FF"                                     NF
·340 PRINT"[HOME]"                       OO
·490 GOSUB 50000                         IM
·500 W$="WINDOW COMMANDS[3"[EP]"]OPEN  -
 SYS"+STR$(OW)+"[EP]WRITE - SYS"+STR$(WW) EG
·510 W$=W$+"[EP]CLOSE - SYS"+STR$(CW)    HD
·520 W%=20:H%=10:R%=5:C%=10              IB
·530 SYS 49432:FORD=1TO2000:NEXT         KE
·540 SYS 49440:FORD=1TO2000:NEXT:GOTO530 PL
·600 END                                 IC
·50000 ML=49408:REM * CAN BE RELOCATED TO
 ANY 256 BLOCK INCREMENT.                BE
·50010 HB=INT(ML/256):LB=(ML/256-HB)*256 IM
·50020 FORN=0TO510:READA:POKEML+N,A:NEXT GK
·50030 POKEML+15,LB:POKEML+16,HB:OW=ML+24
 :WW=ML+28:CW=ML+32:RETURN               NE
·50070 DATA87,215,200,210,195,157,0,197,2
 32,96,189,0,197,96,189,0,193,149,112,202 HK
·50080 DATA16,248,48,26,169,0,240,6,169,6
 4,208,2,169,128,133,252,162,64,181,112,1
 57                                      JP
·50090 DATA176,2,202,208,248,162,17,208,2
 20,162,4,160,0,165,46,133,72,165,45,133,
 71                                      NK
·50100 DATA165,72,197,48,208,10,165,71,19
 7,47,208,4,169,0,240,33,181,112,209,71,2
 08                                      GD
·50110 DATA8,169,128,200,209,71,240,12,13
 6,165,71,24,105,7,144,217,230,72,208,213 LL
·50120 DATA224,0,240,20,200,200,177,71,22
 4,3,176,10,201,3,176,6,169,3,224,0,240,2 FK
·50130 DATA149,153,202,16,178,166,155,134
 ,158,138,202,202,134,159,170,169,0,24,10
 1                                       KH
·50140 DATA154,176,37,202,208,248,133,163
 ,166,154,202,202,202,134,160,169,40,56,2
 29                                      BK
·50150 DATA154,144,18,197,157,144,14,133,
 253,160,0,166,156,138,24,101,155,201,26 HM
·50160 DATA144,11,162,64,189,176,2,149,11
 2,202,16,248,96,165,157,166,156,240,9,24 LD
·50170 DATA105,40,144,1,200,202,208,247,1
 33,164,133,166,133,168,133,170,152,24,10
 5                                       EM
·50180 DATA4,133,165,133,169,105,212,133,
 167,133,171,165,127,133,123,133,118,36,2
 52                                      OL
·50190 DATA48,63,112,61,166,128,232,232,1
 34,119,162,0,160,0,165,154,133,172,177,1
 64                                      KN
·50200 DATA32,117,0,177,166,32,117,0,208,
 2,230,119,200,208,4,230,165,230,167,198 EH
·50210 DATA172,208,231,198,158,240,14,152
 ,24,101,253,144,4,230,165,230,167,168,24 JD
·50220 DATA144,209,36,252,48,118,166,128,
 232,232,232,232,134,119,160,4,162,3,177 CC
·50230 DATA71,149,172,136,202,208,248,136
 ,169,112,32,117,0,165,160,133,172,169,64 FG
·50240 DATA32,117,0,198,172,16,249,192,0,
 208,70,169,110,32,117,0,169,93,32,117,0 GM
·50250 DATA165,160,133,172,177,174,201,64
 ,144,6,201,96,176,2,233,63,198,173,16,4 BP
·50260 DATA230,173,169,32,201,28,208,8,13
 6,165,172,208,243,200,208,242,32,117,0,2
```

To Enter DISINTEGRATOR...

you must use our *Flankspeed* machine language entry program. Read the instructions for *Flankspeed* on page 88.

```
00                                          CA
·50270 DATA198,172,16,216,169,93,32,117,0
,198,159,208,198,169,109,32,117,0,208,16
9                                           GL
·50280 DATA169,125,32,117,0,166,128,232,2
32,36,252,48,2,232,232,134,124,162,0,160 JL
·50290 DATA0,165,154,133,172,32,122,0,145
,168,36,252,48,8,173,134,2,145,170,24,14
4                                           CE
·50300 DATA6,232,32,122,0,145,170,232,208
,2,230,124,200,208,4,230,169,230,171,198 LB
·50310 DATA172,208,217,198,155,208,11,162
,64,189,176,2,149,112,202,16,248,96,152  OJ
·50320 DATA24,101,253,144,4,230,169,230,1
71,168,24,144,184                           JF
```

VIC 20 VERSION

```
·0 GOTO50                                   PD
·1 HB=PEEK(56):LB=PEEK(55)                  ON
·2 HB=HB-1:LB=LB-162:IFLB<0THENLB=LB+256    MG
·3 POKE56,HB:POKE52,HB:POKE55,LB:POKE51,L
B:ML=256*HB+LB                              GB
·4 FORN=0TO417:READA:POKEML+N,A:NEXT        FC
·5 POKEML+22,LB:POKEML+23,HB:OW=ML+5:WW=M
L+9:CW=ML+13                                NB
·6 W$="OPEN  SYS"+STR$(OW)+"[EP]WRITE SYS
"+STR$(WW)+"[EP]CLOSE SYS"+STR$(CW)         EE
·8 W%=18:H%=5:R%=8:C%=2                     JF
·9 RETURN                                   IM
·10 DATA87,215,200,210,195,169,0,240,6,16
9,64,208,2,169,128,133,94,162,4,160,0,18
9,0                                         MH
·11 DATA28,133,87,165,46,133,72,165,45,13
3,71,165,72,197,48,208,10,165,71,197,47,
208                                         JE
·12 DATA4,169,0,240,33,165,87,209,71,208,
8,169,128,200,209,71,240,12,136,165,71,2
4,105                                       BD
·13 DATA7,144,217,230,72,208,213,224,0,24
0,20,200,200,177,71,224,3,176,10,201,3,1
76,6                                        LC
·14 DATA169,3,224,0,240,2,149,87,202,16,1
73,166,89,224,13,176,15,202,202,134,92,1
66,89                                       MB
·15 DATA169,0,24,101,88,201,96,144,3,96,2
40,139,202,208,243,133,96,166,88,202,202
,202                                        KL
·16 DATA134,93,169,22,56,229,88,144,234,1
97,91,144,230,133,95,160,0,166,90,138,24
,101                                        AG
·17 DATA89,201,24,176,216,165,91,166,90,2
40,9,24,105,22,144,1,200,202,208,247,133
,38                                         HA
·18 DATA133,40,152,24,109,136,2,133,39,15
2,24,105,148,44,2,144,16,3,24,105,2,133,
41,36                                       HP
·19 DATA94,48,37,112,37,162,0,160,0,165,8
8,133,42,177,38,157,52,3,177,40,157,152,
3,232                                       FD
·20 DATA200,198,42,208,240,198,89,240,143
,152,24,101,95,168,208,225,208,117,160,4
,162                                        GH
·21 DATA3,177,71,149,251,136,202,208,248,
136,169,112,157,161,2,232,165,93,133,42,
169                                         LE
·22 DATA64,157,161,2,232,198,42,16,248,19
2,0,208,75,169,110,157,161,2,232,169,93,
157                                         HJ
·23 DATA161,2,232,165,93,133,42,177,253,2
01,64,144,6,201,96,176,2,233,63,198,252,
16,4                                        DL
·24 DATA230,252,169,32,201,28,208,8,136,1
65,42,208,243,200,208,242,157,161,2,232,
200                                         PC
·25 DATA198,42,16,215,169,93,157,161,2,23
2,198,92,208,195,169,109,157,161,2,232,2
08,163                                      FF
·26 DATA169,125,157,161,2,162,0,160,0,165
,88,133,42,36,94,16,14,189,52,3,145,38,1
89,152                                      IP
·27 DATA3,145,40,232,200,208,12,189,161,2
,145,38,173,134,2,145,40,232,200,198,42,
208                                         KL
·28 DATA222,198,89,208,1,96,152,24,101,95
,168,24,144,205                             CL
·50 PRINT"[CLEAR]":POKE36879,232            CE
·60 PRINT"[UP][RVSON][YELLOW][22" "]"       CD
·70 FORA=1TO17:PRINT"[YELLOW][RVSON][UP][
4" "][RVSOFF][BLACK]  WINDOW VIC  [YELLO
W][RVSON][4" "]":NEXT                       EE
·75 PRINTTAB(3)"[WHITE]BY THOMAS BUNKER":
PRINTTAB(4)"& AHOY! STAFF"                  IH
·80 GOSUB 1:POKE646,2                       JK
·90 SYS OW:FORD=1TO2000:NEXT                FK
·100 SYS CW:FORD=1TO2000:NEXT:GOTO90        JC
```

DISINTEGRATOR

FROM PAGE 17

First byte: C000   Last byte: C720   SYS to Start: 49152

```
C000:  A9 00 8D 21 D0 A9 0B 8D 6B
C008:  20 D0 A9 93 20 D2 FF 20 49
C010:  4E C4 A0 00 A9 07 99 68 76
C018:  D9 C8 C0 28 D0 F6 A0 00 0C
C020:  B9 D5 C6 99 6E 05 C9 20 6D
C028:  F0 13 8C C6 C6 A9 11 8D 8E
C030:  D3 C6 20 23 C4 A2 64 20 F9
C038:  B2 C3 AC C6 C6 C8 C0 19 8B
C040:  D0 DE A9 21 8D D3 C6 20 03
C048:  C3 C3 A0 00 B9 E6 C5 99 70
C050:  40 03 C8 C0 C0 D0 F5 A9 4E
C058:  0D 8D F8 07 A9 0E 8D F9 32
```

```
C060: 07 A9 0F 8D FA 07 A9 08 61
C068: 8D 29 D0 A9 01 8D 1C D0 15
C070: A9 02 8D 27 D0 A9 0D 8D E5
C078: 25 D0 A9 06 8D 26 D0 A9 4C
C080: 07 8D 28 D0 A9 00 8D 10 55
C088: D0 A9 01 8D 17 D0 A9 B1 D4
C090: 8D C7 C6 A9 B3 8D CC C6 2B
C098: A9 09 8D CF C6 A9 00 8D A6
C0A0: D4 C6 A9 B0 8D C8 C6 8D 41
C0A8: C9 C6 8D CA C6 8D CB C6 78
C0B0: A9 00 8D D0 C6 A9 EE 85 9D
C0B8: FB A9 C6 85 FC 20 34 C3 BF
C0C0: 20 E4 FF C9 0D D0 F9 A9 11
C0C8: 63 85 FB A9 C4 85 FC 20 BE
C0D0: 34 C3 A9 77 85 FB A9 C4 D9
C0D8: 85 FC 20 34 C3 A9 C2 85 65
C0E0: FB A9 C4 85 FC 20 34 C3 E5
C0E8: A9 ED 85 FB A9 C4 85 FC F2
C0F0: 20 34 C3 A9 C2 85 FB A9 A0
C0F8: C4 85 FC 20 34 C3 A9 28 2A
C100: 85 FB A9 C5 85 FC 20 34 C7
C108: C3 A9 53 85 FB A9 C5 85 3F
C110: FC 20 34 C3 A0 00 A9 A0 10
C118: 99 C0 07 A9 0B 99 C0 DB 64
C120: C8 C0 28 D0 F1 AD C7 C6 D0
C128: 8D 9F 07 AD CC C6 8D BF EA
C130: 07 A9 00 8D 15 D0 20 42 B6
C138: C3 B9 A6 C6 8D 00 D0 B9 3B
C140: AE C6 8D 10 D0 A9 32 8D 8D
C148: 01 D0 A9 01 8D 15 D0 8C C4
C150: C6 C6 20 E4 FF 8D CD C6 05
C158: AE CE C6 A9 07 9D D2 DA 98
C160: 8E CE C6 AD 1F D0 20 BB FD
C168: C3 AE CF C6 20 B2 C3 AE B6
C170: CE C6 A9 0E 9D D2 DA 8E 97
C178: CE C6 AD 1F D0 29 01 F0 C6
C180: 03 4C F0 C2 AD CD C6 C9 8F
C188: 20 D0 03 20 5B C4 C9 31 B7
C190: 90 C0 C9 39 B0 BC AC C6 C5
C198: C6 D9 BE C6 D0 B4 B9 A6 A4
C1A0: C6 8D 02 D0 B9 B6 C6 8D 8C
C1A8: 10 D0 A9 C2 8D 03 D0 A9 01
C1B0: 03 8D 15 D0 20 BB C3 20 E6
C1B8: BF C3 AD 1E D0 AE CF C6 1E
C1C0: 20 B2 C3 AD 1E D0 29 01 1E
C1C8: F0 EA AD 01 D0 8D 05 D0 87
C1D0: AD 00 D0 8D 04 D0 A9 04 5F
C1D8: 8D 15 D0 A9 81 8D D3 C6 9F
C1E0: 20 23 C4 EE C8 C6 AD C8 DD
C1E8: C6 C9 BA F0 06 8D AF 07 6F
C1F0: 4C 34 C2 A9 B0 8D C8 C6 AB
C1F8: AD C8 C6 8D AF 07 EE C9 33
C200: C6 AD C9 C6 C9 BA F0 06 80
C208: 8D AE 07 4C 34 C2 A9 B0 E8
C210: 8D C9 C6 AD C9 C6 8D AE A8
C218: 07 EE CA C6 AD CA C6 C9 A8
C220: BA F0 06 8D AD 07 4C 34 94
C228: C2 A9 B0 8D CA C6 AD CA DC
C230: C6 8D AD 07 EE D0 C6 AD 6D
C238: D0 C6 C9 0F F0 03 4C 31 1A
C240: C1 A9 00 8D D0 C6 A9 00 7A
C248: 8D 15 D0 EE C7 C6 AD C7 AE
C250: C6 8D 9F 07 A9 A2 85 FB 19
C258: A9 C5 85 FC 20 34 C3 A9 0C
C260: 11 8D D3 C6 20 C3 C3 A2 E3
C268: FF 20 B2 C3 A2 FF 20 B2 74
C270: C3 A2 FF 20 B2 C3 A2 FF 10
C278: 20 B2 C3 A9 C5 85 FB A9 A9
C280: C5 85 FC 20 34 C3 CE CF 7F
C288: C6 F0 03 4C 31 C1 A9 00 2C
C290: 8D 15 D0 A9 98 85 FB A9 71
C298: C5 85 FC 20 34 C3 AD CB 72
C2A0: C6 49 80 8D 06 04 AD CA 41
C2A8: C6 49 80 8D 07 04 AD C9 49
C2B0: C6 49 80 8D 08 04 AD C8 51
C2B8: C6 49 80 8D 09 04 A9 30 BD
C2C0: 8D 0A 04 A0 00 A9 01 99 41
C2C8: 06 D8 C8 C0 05 D0 F6 A9 A7
C2D0: 0D 20 D2 FF A9 84 85 FB 80
C2D8: A9 C5 85 FC 20 34 C3 20 03
C2E0: E4 FF C9 59 F0 07 C9 4E F8
C2E8: D0 F5 20 74 A4 4C 8E C0 84
C2F0: A9 81 8D D3 C6 EE D4 C6 CE
C2F8: 20 C3 C3 CE D4 C6 CE CC A6
C300: C6 AD CC C6 C9 B0 F0 11 84
C308: AD CC C6 8D BF 07 A9 00 47
C310: 8D 15 D0 AD 1F D0 4C 31 9E
C318: C1 A9 93 20 D2 FF A9 00 B3
C320: 8D 15 D0 A9 00 8D D3 C6 65
C328: EE D4 C6 20 C3 C3 CE D4 FD
C330: C6 4C 8E C2 A0 00 B1 FB E2
C338: F0 07 20 D2 FF C8 4C 36 6E
C340: C3 60 20 97 E0 A5 8E C9 FA
C348: 20 B0 05 A0 00 4C 88 C3 57
C350: C9 40 B0 05 A0 01 4C 88 86
C358: C3 C9 60 B0 05 A0 02 4C EA
C360: 88 C3 C9 80 B0 05 A0 03 50
C368: 4C 88 C3 C9 A0 B0 05 A0 C1
C370: 04 4C 88 C3 C9 C0 B0 05 4D
C378: A0 05 4C 88 C3 C9 E0 B0 12
C380: 05 A0 06 4C 88 C3 A0 07 6C
C388: 8C D1 C6 AD D2 C6 CD D1 94
C390: C6 F0 AF AD D1 C6 8D D2 9E
C398: C6 8C C6 C6 A2 00 AC C6 8F
C3A0: C6 F0 08 E8 E8 E8 E8 E8 EC
C3A8: 88 D0 F8 8E CE C6 AC C6 92
C3B0: C6 60 A0 00 C8 D0 FD CA DA
C3B8: D0 F8 60 EE 01 D0 60 CE D2
C3C0: 03 D0 60 20 4E C4 A9 0F E0
C3C8: 8D 18 D4 A9 F0 8D 06 D4 46
C3D0: A9 0E 8D D1 C6 AD D3 C6 F6
C3D8: 8D 04 D4 A9 19 8D 01 D4 65
C3E0: A9 B1 8D 00 D4 AD D1 C6 E4
C3E8: 8D 18 D4 A2 32 20 B2 C3 CE
C3F0: AD D1 C6 8D 20 D0 AD D4 38
C3F8: C6 F0 06 AD D1 C6 8D 21 AB
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 87 and 88 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
C400: D0 CE D3 C6 AD D3 C6 8D 10
C408: 04 D4 EE D3 C6 CE D1 C6 D1
C410: AD D1 C6 D0 C0 20 4E C4 1B
C418: A9 0B 8D 20 D0 A9 00 8D 82
C420: 21 D0 60 20 4E C4 A9 0F 5E
C428: 8D 18 D4 A9 70 8D 05 D4 24
C430: A9 FA 8D 06 D4 AD D3 C6 85
C438: 8D 04 D4 A9 09 8D 01 D4 B4
C440: A9 9F 8D 00 D4 A2 C8 20 77
C448: B2 C3 20 4E C4 60 A0 00 F2
C450: A9 00 99 00 D4 C8 C0 19 0B
C458: D0 F6 60 20 E4 FF C9 20 6F
C460: D0 F9 60 93 11 11 11 11 63
C468: 11 11 11 11 11 11 11 11 F0
C470: 11 11 11 11 11 11 00 12 E8
C478: 1C 20 20 92 9A C1 12 1C F1
C480: 20 20 20 20 92 9A C1 12 02
C488: 1C 20 20 20 20 92 9A C1 14
C490: 12 1C 20 20 20 20 92 9A 6C
C498: C1 12 1C 20 20 20 20 92 9B
C4A0: 9A C1 12 1C 20 20 20 20 AB
C4A8: 92 9A C1 12 1C 20 20 20 26
C4B0: 20 92 9A C1 12 1C 20 20 2E
C4B8: 20 20 92 9A C1 12 1C 20 36
C4C0: 20 00 12 1C 20 20 20 20 8F
C4C8: 20 20 20 20 20 20 20 20 C9
C4D0: 20 20 20 20 20 20 20 20 D1
C4D8: 20 20 20 20 20 20 20 20 D9
C4E0: 20 20 20 20 20 20 20 20 E1
C4E8: 20 20 20 20 00 12 1C 20 B7
C4F0: 20 05 31 1C 20 20 20 20 E3
C4F8: 05 32 1C 20 20 20 20 05 D1
C500: 33 1C 20 20 20 20 05 34 09
C508: 1C 20 20 20 20 05 35 1C FA
C510: 20 20 20 20 05 36 1C 20 08
C518: 20 20 20 05 37 1C 20 20 11
C520: 20 20 05 38 1C 20 20 00 F9
C528: 12 9A 20 20 20 20 20 20 95
C530: 20 20 20 20 20 20 20 20 31
C538: 20 20 20 20 20 20 20 20 39
C540: 20 20 20 20 20 20 20 20 41
C548: 20 20 20 20 20 20 20 20 49
C550: 20 20 00 12 99 4C 45 56 24
C558: 45 4C 20 30 30 9A 20 20 45
C560: 20 20 20 20 99 53 43 4F 60
C568: 52 45 20 30 30 30 30 30 11
C570: 9A 20 20 20 20 20 20 20 EB
C578: 99 4C 49 56 45 53 20 30 E6
C580: 30 13 05 00 11 11 50 4C 87
C588: 41 59 20 41 47 41 49 4E A4
C590: 3F 20 5B 59 2F 4E 5D 00 7F
C598: 93 05 92 53 43 4F 52 45 41
C5A0: 20 00 13 05 92 11 11 11 9E
C5A8: 11 1D 1D 1D 1D 1D 1D 1D 85
C5B0: 1D 1D 1D 1D 1D 1D 4C 45 F0
C5B8: 56 45 4C 20 43 4F 4D 50 F0
C5C0: 4C 45 54 45 00 13 11 11 21
C5C8: 11 11 1D 1D 1D 1D 1D 1D 99
C5D0: 1D 1D 1D 1D 1D 1D 1D 20 BC
C5D8: 20 20 20 20 20 20 20 20 D9
C5E0: 20 20 20 20 20 00 0A 02 8D
C5E8: 80 0A 02 80 4A 02 84 FA C1
C5F0: 02 BC FA 02 BC FF 03 FC 69
C5F8: DF CF DC D7 CF 5C F5 FD 7D
C600: 7C 3F FF F0 0F 57 C0 0F E2
C608: 57 C0 0F 57 C0 0F 57 C0 6E
C610: 0F 57 C0 03 FF 00 00 FC 37
C618: 00 00 10 00 00 10 00 00 38
C620: 00 00 00 00 00 00 00 10 30
C628: 00 00 38 00 00 7C 00 00 DC
C630: FE 00 00 FE 00 00 38 00 66
C638: 00 7C 00 00 00 00 00 00 B4
C640: 00 00 00 00 00 00 00 00 40
C648: 00 00 00 00 00 00 00 00 48
C650: 00 00 00 00 00 00 00 00 50
C658: 00 00 00 00 00 00 00 00 58
C660: 00 00 00 00 00 00 0E 01 6F
C668: FC 1F A3 FC 3F FF F8 7F DC
C670: FF 80 7F FF C0 FF FF F0 22
C678: FF FF F8 7F FF FC 3F FF 2D
C680: FC 3F FF FC 1F FF F2 1F EA
C688: FF F0 3F FF F8 7F FF FC 2E
C690: 7F FF FE 3F FF FF 3F FF 8D
C698: FF 7F FF FF FF 9F FE FC B3
C6A0: 0F 9E F8 07 0E FF 20 48 C4
C6A8: 70 98 C0 E8 11 39 00 00 A5
C6B0: 00 00 00 00 01 01 00 00 B2
C6B8: 00 00 00 00 07 07 31 32 2A
C6C0: 33 34 35 36 37 38 00 00 03
C6C8: 00 00 00 00 00 00 00 00 C8
C6D0: 00 00 00 00 00 04 20 09 FD
C6D8: 20 13 20 09 20 0E 20 14 97
C6E0: 20 05 20 07 20 12 20 01 80
C6E8: 20 14 20 0F 20 12 13 11 A2
C6F0: 11 11 11 11 11 11 11 11 79
C6F8: 11 11 11 11 11 11 11 1D 8D
C700: 1D 1D 1D 1D 1D 1D 1D 9A 66
C708: 50 52 45 53 53 20 12 52 1B
C710: 45 54 55 52 4E 92 20 54 A6
C718: 4F 20 53 54 41 52 54 00 17
C720: FF 20
```

SCREEN DUMP ROUTINE

FROM PAGE 46

```
·49000 PRINT"[CLEAR]":POKE53280,9:POKE532
 81,7                                        JJ
·49010 FORA=1TO5:PRINT:NEXT                  BF
·49020 PRINTTAB(10)"[c 2]SCREEN DUMP UTIL
 ITY"                                        PO
·49030 PRINTTAB(19)"[DOWN]BY":PRINTTAB(12
 )"[DOWN]ALFRED J. BRUEY"                    OP
```

```
·49040 PRINTTAB(12)"[7"[DOWN]"][BLACK]FOR
  AHOY MAGAZINE"                          CJ
·49050 PRINTTAB(7)"[DOWN](C) ION INTERNAT
IONAL INC."                               MD
·49060 PRINTTAB(5)"[DOWN][DOWN][PURPLE][R
VSON]PRESS 'F1' TO DUMP THIS SCREEN"      LG
·49070 GETA$:IFA$<>"[F1]"THEN49070        DA
·49080 GOSUB 50030                        IL
·49090 PRINT"[CLEAR][WHITE]":POKE53280,0:
POKE53281,6:FORA=1TO8:PRINT:NEXT          EL
·49100 PRINT"  IF YOU WISH TO USE THIS UT
ILITY AS A  SUBROUTINE IN YOUR OWN"       AG
·49110 PRINTTAB(24)"[UP]PROGRAM, REMOVE L
INES 49[3"0"] TO 50030 AND THEN ADD THE"  LE
·49120 PRINT"REMAINING LINES TO YOUR OWN
PROGRAM."                                 KC
·49130 END                                IC
·50000 :                                  DI
·50010 REM ** SCREEN DUMP SUBROUTINE **   EL
·50020 :                                  DI
·50030 OPEN4,4                            DG
·50040 FORI=1023 TO 2006 STEP 40          OK
·50050 PR$=""                             LM
·50060 FOR J=I+1 TO I+40                  KB
·50070 V=PEEK(J)                          GD
·50080 IF V>127 THEN 50130                NH
·50090 IFV<=31 OR V>=96 THEN V1=V+64:GOTO
 50120                                    AG
·50100 IF V>=32 AND V<=63 THEN V1=V:GOTO
 50120                                    KC
·50110 V1=V+128                           OB
·50120 PR$=PR$+CHR$(V1):GOTO 50190        GC
·50130 IF V<=159 OR V>=224 THEN V1=V-64:G
OTO 50160                                 FK
·50140 IF V>=192 AND V<=223 THEN V1=V:GOT
O 50160                                   CC
·50150 V1=V-128                           NA
·50160 IF LEN(PR$)=0 THEN PR$=CHR$(18)+CH
R$(V1)+CHR$(146):GOTO 50190               FM
·50170 IFRIGHT$(PR$,1)<>CHR$(146)THENPR$=
PR$+CHR$(18)+CHR$(V1)+CHR$(146):GOTO5019
0                                         FE
·50180 PR$=LEFT$(PR$,LEN(PR$)-1)+CHR$(V1)
+CHR$(146)                                NO
·50190 NEXTJ                              MM
·50200 PRINT#4,PR$                        MG
·50210 NEXTI:CLOSE4                       OH
·50220 RETURN                             IM
```

FORMATTER

FROM PAGE 13

```
·10 REM *** FORMATTER ***                 FE
·15 IF PEEK(686)THEN386                   LH
·20 GOTO1000                              FC
·95 REM ***FLIP SCREENS***                MP
·100 X=PEEK(T):IFP<>ETHEN125              JP
·102 SYS49391                             KI
·105 P=I:POKEA,49:POKET+B,P:POKE4,76:S=H  LA
·106 POKE56333,127:POKE788,196:POKE789,19
4:POKEA+B,129:REM START RASTER            NC
·107 PRINT"[CLEAR][DOWN][RIGHT][RVSOFF][B
LACK]TYPE CHARACTER TO":PRINT"[RIGHT]REP
ROGRAM ON LOWER                           CL
·109 PRINT"[RIGHT]PORTION OF SCREEN AND":
PRINT"[RIGHT]PRESS F4. PRESS ANY F        HM
·111 PRINT"[RIGHT]KEY TO INPUT CHANGES.":
PRINT"[DOWN][RIGHT]PRESS F6 TO RETURN     GG
·113 PRINT"[RIGHT]TO SCREEN 1[DOWN]":POKE
T,X:RETURN                                FK
·124 REM SCREEN @ BANK 2, 34816($8800)AND
 CHARS @ $8000 OR $9000                   PG
·125 POKEA+B,240:POKE788,49:POKE789,234:P
OKE56333,129:REM RASTER OFF               LJ
·126 P=E:SYS49424:POKER+H,SC:POKER,BC:POK
EA,CS:S=.                                 CC
·127 POKET+B,P:POKE4,80:POKET,X:PRINT"[HO
ME]":RETURN                               JH
·149 REM **REPROGRAM***                   EE
·150 G=PEEK(3):X=PEEK(T)                  BK
·151 FORJ=.TO5:POKE218+J,136:POKE224+J,13
7:NEXT                                    JO
```

```
•152 M=32768+G*C:PRINT"[HOME]"TAB(27)"[11
" "][HOME]"                              IC
•153 FORJ=.TO7:Z=128:G=PEEK(M+J):PRINTTAB
(27)"[3" "]";                            LF
•154 FORK=HTOC:PRINT"[CYAN]0";:IFG>=ZTHEN
G=G-Z:PRINT"[RED][LEFT]*";               NN
•155 Z=Z/B:NEXT:PRINT" ":NEXT:PRINTTAB(27
)"[13" "]"                               IA
•156 PRINT"[HOME][DOWN]"TAB(30);         EP
•157 POKEW,.:SYS50222:POKET,X:RETURN     ME
•200 SC=-SC*(SC<N)+H:POKER+H,SC:POKE254,S
C:RETURN:REM SCREEN COLOR                NJ
•225 BC=-BC*(BC<N)+H:POKER,BC:RETURN:REM
BORDER COLOR                             IE
•250 B1=-B1*(B1<N)+H:R1=B1-H:POKER+B,R1:R
ETURN:REM REGISTER 1                     BK
•275 B2=-B2*(B2<N)+H:R2=B2-H:POKER+3,R2:R
ETURN:REM REGISTER 2                     NC
•300 B3=-B3*(B3<N)+H:R3=B3-H:POKER+4,R3:R
ETURN:REM REGISTER 3                     JO
•325 EB=.:POKE53265,PEEK(53265)OR64:REM E
XTENDED BACKGROUND ON                    JE
•326 SYS49980:F=PEEK(253)-132:ONFGOSUB200
,250,275,300,350,380,100                 KE
•328 IFF=5THENRETURN                     PJ
•329 IFF=CTHENEB=H:GOTO400               CN
•330 GOTO326                             CL
•350 POKE53265,PEEK(53265)AND191:RETURN:R
EM EXTENDED BACKGROUND OFF               JB
•365 MC=.:POKEA-B,PEEK(A-B)ORN:REM MULTIC
OLOR ON                                  AC
•367 SYS49980:F=PEEK(253)-132:ONFGOSUB200
,250,275,375,328,380,100                 JL
•370 IFF=4THENRETURN                     PE
•372 IFF=CTHENMC=H:GOTO400               FM
•373 GOTO367                             CO
•375 POKE53270,PEEK(53270)AND239:RETURN:R
EMMULTICOLOR OFF                         FO
•380 IFSTHENGOSUB150:RETURN:REM CHARACTER
 SET SWITCH                              MO
•381 O=ABS(O-CN):CS=33+O:POKEA,CS:RETURN:
REM FLIP CHARACTER SETS                  KG
•385 REM ***MENU***                      OE
•386 P=140:E=136:I=P:R=53280:H=1:BC=H:SC=
15:A=53272:CS=33:T=646:B=2               DN
•388 X=11:W=198:C=8:N=16:CN=4:BL=53265:PO
KE254,SC                                 BB
•390 D=3002                              MJ
•395 POKE56576,5:IFPEEK(684)=.THENGOSUB12
6:PRINT"[CLEAR]";:POKE684,H              AL
•400 X=PEEK(T):IFP=ETHENSYS49391:CS=PEEK(
A)                                       AM
•401 IFSTHENPOKEA+B,240:POKE788,49:POKE78
9,234:POKE56333,129:REMTURN OFF RASTER   LO
•402 IFMCTHENGOSUB375:MC=H               LA
•403 IFEBTHENGOSUB350:EB=H               JB
•408 POKE56576,7:POKET+B,4:REM DEFAULT SC
REEN                                     FD
•410 POKEBL,PEEK(BL)AND239:POKEA,21:POKER
+H,15:POKER,H:POKET,0                    HD
•411 PRINT"[CLEAR]"TAB(57)"MENU":PRINT"[D
OWN][3"[RIGHT]"]1 INSTRUCTION SCREENS    BH
•413 PRINT"[DOWN][3"[RIGHT]"]2 FORMATTING
 SCREEN                                  EI
•414 PRINT"[DOWN][3"[RIGHT]"]3 CHARACTER
REPROGRAMMING SCREEN                     LM
•415 PRINT"[DOWN][3"[RIGHT]"]4 MAKE BASIC
 PRINT STATEMENTS                        MH
•416 PRINT"[DOWN][3"[RIGHT]"]5 DISPLAY NE
W SCREEN(S)                              PC
•417 PRINT"[DOWN][3"[RIGHT]"]6 DELETE PRO
GRAM LINES                               PB
•418 PRINT"[DOWN][3"[RIGHT]"]7 MAKE CHARA
CTER DATA STATEMENTS                     OD
•419 PRINT"[DOWN][3"[RIGHT]"]8 SWITCH CUS
TOM CHARACTER SET":PRINT"[DOWN][3"[RIGHT
"]9 STOP PROGRAM                         FP
•420 PRINTTAB(50)"[BLUE]YOUR CHOICE [1-9]
[SS]?":POKEBL,27                         EH
•421 WAITW,H:G=PEEK(631)-48:POKEW,.:IFG<H
ORG>9THEN421                             DA
•422 ONGGOTO600,430,435,485,1600,720,800,
950,428                                  AM
•428 POKE828,.:SYS828                    DE
•430 POKE56576,5:GOSUB126:GOTO438        OD
•435 POKE56576,5:POKER,BC:GOSUB105       IN
•438 IFMCTHENGOSUB365                    PG
•440 IFEBTHENGOSUB325                    PF
•445 SYS49980:F=PEEK(253)-132            BH
•450 ONFGOSUB200,225,275,365,325,380,100 KP
•455 IFF=CTHEN400                        AM
•460 GOTO445                             CL
•480 REM ***SCREEN SCAN***               NF
•485 PRINT"[CLEAR]"TAB(92)"NEW PRINT LINE
S                                        CL
•490 PRINTTAB(205)". . .SPACE TO CONTINUE
"SPC(58)". . . ANY OTHER FOR MENU        KL
•495 POKEW,.:WAIT198,H:IFPEEK(631)<>32THE
N410                                     MA
•500 PRINT"[CLEAR]"D"GOSUB6[3"0"]2:PRINTC
HR$(19)TAB(140)PEEK(58)*256+PEEK(57)";   PM
•501 PRINT":WAITW,H:POKEW,.":POKE679,X   KI
•502 D=D+B:PRINTD"POKER+H,"SC"[LEFT]:POKE
R,"BC"[LEFT]:POKEA,"CS"[LEFT]:SC="SC"[LE
FT]:BC="BC;                              JB
•503 PRINT"[LEFT]:POKET,"X"[LEFT]:PRINTCH
R$(147);":SYS49457:D=D+B:M=4             EP
•504 IFMCTHENPRINTD"POKE53270,PEEK(53270)
OR16:POKER+2,"R1"[LEFT]:POKER+3,"R2;     PJ
•505 IFMCTHENPRINT"[LEFT]:MC=H":D=D+B    NI
•506 IFEBTHENPRINTD"POKE53265,PEEK(53265)
OR64:POKER+2,"R1"[LEFT]:POKER+3,"R2;     EJ
•507 IFEBTHENPRINT"[LEFT]:POKER+4,"R3"[LE
FT]:EB=H":D=D+B                          MH
•508 PRINT"515 D="D"[LEFT]:EB="EB"[LEFT]:
MC="MC"[LEFT]:X="X                       JD
```

```
•509 IFEBORMCTHENPRINT"PRINTCHR$(147):POK
E218,4:RUN515":M=5:GOTO560                KF
•510 PRINT"RUN515":GOTO560                NB
•517 IFPEEK(2)THENPRINT"[CLEAR]";:POKE218
,4                                        OC
•520 PRINTD"?"CHR$(34):D=D+2              DG
•525 SYS49482:IFPEEK(2)=.THENM=5:GOTO540  OH
•530 PRINT"515 D="D"[LEFT]:EB="EB"[LEFT]:
MC="MC"[LEFT]:X="X                        JD
•535 PRINT"RUN515":M=6:GOTO560            DC
•540 PRINTD"WAITW,H:G=P[s E](631):P[s O]W
,.:IFG=82THENP[s O]T,"X;                  FI
•541 PRINT"[LEFT]:P[s O]254,SC:GOTO438":D
=D+2                                      KF
•545 IFMCTHENPRINTD"POKE53270,PEEK(53270)
AND239:MC=.":D=D+2:M=M+1                  HK
•550 IFEBTHENPRINTD"POKE53265,PEEK(53265)
AND191:EB=.":D=D+2:M=M+1                  DH
•552 PRINTD"IFG<>32THENSC=M:BC=Z:?CHR$(14
7):GOTO408":D=D+2                         IF
•555 PRINT"390 D="D:PRINT"515":PRINT"RUN3
86":IFPEEK(681)THENM=M+1                  BP
•560 F=631:G=13:POKEF,19:FORJ=1TOM:POKEF+
J,G:NEXT:POKE198,M+1:END                  DD
•595 REM ***INSTRUCTIONS***               IE
•600 POKEBL,PEEK(BL)AND239:POKER+H,H:POKE
R,6:POKEA,23                              AJ
•602 PRINT"[CLEAR][BLUE][3" "][s O][s N][
SS][s F][s O][s R][s M][s A][s T][SS][s
A][s N][s D][SS][s R][s E][s P][s R][s O
][s G][s R][s A][s M][SS][s S][s C][s R]
[s E][s E][s N][s S][SS]-                 EC
•604 PRINTTAB(41)"F1 = [s S]CREEN COLOR   LF
•606 PRINTTAB(41)"F2 = [s E]XTENDED BACKG
ROUND COLOR ON/OF                         PM
•608 PRINTTAB(41)"F3 = [s B]ORDER COLOR O
R COLOR REGISTER #1                       BA
•610 PRINTTAB(6)"WITH MULTICOLOR ([s M][s
 C]) OR EXTENDED"SPC(8)"BACKGROUND COLOR
 ([s E][s B])                             JO
•614 PRINTTAB(41)"F4 = [s S]WITCH BETWEEN
 CUSTOM AND REGULAR                       LO
•616 PRINTTAB(6)"CHARACTERS ON FORMAT SCR
EEN OR                                    AN
•618 PRINTTAB(6)"ENABLE CUSTOM CHARACTERS
 ON"SPC(13)"REPROGRAMMING SCREEN          JH
•620 PRINTTAB(41)"F5 = [s C]OLOR REGISTER
 #2 WITH [s M][s C] OR [s E][s B]"        GM
•622 PRINTTAB(41)"F6 = [s S]WITCH BETWEEN
 FORMATTING AND"SPC(11)"REPROGRAMMING SC
REEN                                      AN
•624 PRINTTAB(41)"F7 = [s M]ULTICOLOR ON/
OFF OR COLOR                              IG
•626 PRINTTAB(6)"REGISTER #3 WITH [s E][s
 B]":POKEBL,27                            JH
•628 PRINT"[DOWN][RIGHT]":FORJ=.TO10:PRIN
T"[UP][RIGHT][RED]F8 = [s M]ENU"          JB
•630 FORG=HTO60:NEXT:PRINT"[UP][RIGHT][RV
SON]F8 = [s M]ENU":FORG=HTO60:NEXT:NEXT   NA
•632 PRINT"[RIGHT][BLUE] ([s S][s P][s A]
[s C][s E] FOR MORE: [s A]NY OTHER FOR M
ENU)";                                    LK
•634 WAITW,H:POKEW,.:IFPEEK(631)=32THEN65
6                                         PP
•636 GOTO410                              CJ
•656 POKEBL,PEEK(BL)AND239:PRINT"[CLEAR]"
TAB(86)"[s F][s O][s R][s M][s A][s T][S
S][s S][s C][s R][s E][s E][s N][SS][s C
][s O][s L][s O][s R][SS][s C][s H][s A]
[s N][s G][s E][s S]"                     GP
•658 PRINTTAB(83)"[s T]O CHANGE ALL OF A
CERTAIN COLOR                             PK
•660 PRINTTAB(3)"TO ANOTHER COLOR:        BN
•662 PRINTTAB(45)"A. [s S]ET CURSOR TO CO
LOR YOU WISH"SPC(12)"TO CHANGE.           MM
•664 PRINTTAB(45)"B. [s P]RESS [s C][s T]
[s R][s L] AND C KEYS                     DH
•666 PRINTTAB(8)"TOGETHER.  ([s T]HE CURS
OR WILL"SPC(13)"DISAPPEAR.)               EJ
•668 PRINTTAB(45)"C. [s P]RESS [s C][s T]
[s R][s L] OR [s C][s O][s M][s M][s O][
s D][s O][s R][s E]                       DI
•670 PRINTTAB(8)"KEY AND THE NEW COLOR KE
Y"SPC(15)"TOGETHER.                       AM
•672 PRINTTAB(50)"([s A]NY KEY FOR MENU.)
":POKEBL,27                               FD
•674 WAITW,H:POKEW,.:GOTO410              CD
•719 REM ***DELETE***                     FA
•720 POKER+H,3 :POKER,B:POKEA,23:POKEBL,P
EEK(BL)AND239                             FC
•722 PRINT"[CLEAR]"TAB(207)"[BLUE][s D][s
 E][s L][s E][s T][s E] [s P][s R][s O][
s G][s R][s A][s M][SS][s P][s R][s I][s
 N][s T] [s L][s I][s N][s E][s S]        MG
•724 PRINTTAB(86)"1 [s A]LL NEW PRINT LIN
ES                                        LI
•726 PRINTTAB(46)"2 [s S]ELECT FROM TO EN
D                                         FD
•728 PRINTTAB(46)"3 [s S]ELECT FROM TO SE
LECT END                                  BP
•729 PRINTTAB(46)"4 [s D]ELETE ALL [s B][
s U][s T] NEW LINES                       DP
•730 PRINTTAB(46)"5 [s R]ETURN TO MENU    FL
•732 PRINTTAB(49)"([s P]RESS NUMBER DESIR
ED)":POKEBL,27                            OI
•734 WAITW,H:G=PEEK(631)-48:POKEW,.:IFG<H
ORG>5THEN734                              DF
•736 IFD=3002THENIFG<4THENPRINTTAB(81)"[R
ED][RVSON][s N]O LINES TO DELETE.  [s A]
NY KEY FOR MENU.":GOTO685                 OA
•738 POKET,X:M=.:F=D:Z=3002:L=Z:ONGGOTO77
3,740,740,900,410                         IF
•740 PRINT"[CLEAR][DOWN][DOWN][RIGHT][RIG
HT][s E]NTER STARTING LINE NUMBER";       DI
•745 INPUTL:IFLAND1ORL<3002THENPRINT"[RIG
HT][RIGHT]# UNEVEN OR TOO SMALL:";:GOTO7
```

```
45                                          HB
•750 IFG=2THENZ=L:GOTO765                   CK
•755 Z=D:PRINT"[DOWN][RIGHT][RIGHT][s E]N
TER ENDING LINE NUMBER";                    AG
•760 INPUTD:IFDAND1ORD>ZTHENPRINT"# UNEVE
N OR TOO BIG:";:GOTO760                     JH
•765 IFL>=DTHENPRINT"[CLEAR][DOWN][4"[RIG
HT]"][RED][RVSON][s S]TARTING # TOO BIG.
 [s S]TART OVER.[RVSOFF][c 4]":D=F:GOTO7
22                                          KN
•773 F=D-L:PRINT"[CLEAR]";:IFF<11THEN785    HB
•775 FORJ=LTOL+12STEP2:PRINTJ:NEXT:         KM
•776 PRINT"772 L="J"[LEFT]:D="D"[LEFT]:Z=
"Z:PRINT"RUN772":M=9:GOTO560                KO
•785 IFF=.THEN787                           AJ
•786 FORJ=LTODSTEP2:PRINTJ:M=M+1:NEXT       LD
•787 PRINT"772":PRINT"390 D="Z:PRINT"RUN3
86":M=M+3:GOTO560                           BA
•795 REM***CHARACTER DATA***                BF
•800 SYS49280:IFPEEK(B)THEN820              PH
•805 PRINT"[CLEAR]"TAB(203)"[c 4]NO CHARA
CTERS REPROGRAMMED AS YET                   FF
•810 PRINT"[DOWN][DOWN][3"[RIGHT]"][c 4](
PRESS ANY KEY TO RETURN TO MENU)            CD
•815 WAITW,H:POKEW,.:POKE254,SC:GOTO410     DD
•820 PRINT"[CLEAR]6[3"0"]6 FORJ=1TO"PEEK(
2):PRINT"RUN825":PRINT"  60010":M=3:GOTO
560                                         OK
•825 INPUTZ                                 OD
•830 SC=PEEK(50432+PEEK(2)-1):CH=32768+8*
SC                                          LI
•835 FORJ=.TO7:C$=C$+STR$(PEEK(CH+J)):NEX
T                                           MH
•840 FORK=2TOLEN(C$):Q$=MID$(C$,K,1):IFQ$
=" "THENQ$=","                              FN
•845 D$=D$+Q$:NEXT                          GL
•850 PRINT"[CLEAR]"Z"DATA"SC"[LEFT],"D$     DE
•855 Z=Z+2:POKE2,PEEK(2)-1:IFPEEK(2)=.THE
NPRINT"RUN386":M=2:GOTO560                  AD
•860 PRINT"RUN825":PRINT"  "Z:M=3:GOTO560   PG
•895 REM ***DELETE OLD***                   DC
•900 POKER+H,3:POKER,B:PRINT"[CLEAR]"TAB(
170)"[BLACK][s D][s E][s L][s E][s T][s
E][SS][s O][s L][s D][SS][s P][s R][s O]
[s G][s R][s A][s M]                        GL
•904 PRINTTAB(42)"[s A]S WRITTEN, NEW SCR
EENS WILL DISPLAY                           OL
•906 PRINT"[RIGHT][RIGHT]ON SCREEN AT 348
16 WITH [s R][s O][s M] IMAGE AT            IP
•908 PRINT"[RIGHT][RIGHT]36864 AND PROGRA
MMABLE CHARACTERS AT"SPC(4)"32768.          AD
•914 PRINTTAB(42)"[s T]O DISPLAY &/OR SAV
E ON DEFAULT                                AI
•916 PRINT"[RIGHT][RIGHT]SCREEN, DELETE L
INE 3[3"0"] AND DELETE                      BJ
•918 PRINT"[RIGHT][RIGHT]OR REVISE [s P]O
KE 53272 VALUES TO POINT                    IC
•920 PRINT"[RIGHT][RIGHT]TO YOUR [s R][s
A][s M] CHARACTER SET.                      NI
•924 PRINTTAB(47)"([s H]IT ANY KEY TO PRO
CEED)                                       NO
•926 PRINTTAB(7)"('[s R]' TO RETURN TO ME
NU.)                                        JA
•928 WAITW,H:G=PEEK(631):POKEW,.:IFG=82TH
EN410                                       AM
•930 SYS49152:POKER+H,10:PRINTTAB"[CLEAR]
"TAB(250)"[RVSON][RED] OLD PROGRAM GONE!
"                                           HB
•935 PRINTTAB(124)"[BLUE][s T]O RECOVER O
LD PROGRAM, ENTER IN"SPC(48)"DIRECT MODE
:                                           JA
•940 PRINT"[DOWN][DOWN][BLACK]POKE43,1:PO
KE641,1:POKE44,8:POKE642,8:RUN":END         BE
•945 REM ***SWITCH CUSTOM SETS***           CG
•950 POKER+H,7:POKER,1:POKEA,23:POKEBL,PE
EK(BL)AND239                                HD
•952 PRINT"[CLEAR][c 4][DOWN]"TAB(251)"[s
 S][s W][s I][s T][s C][s H][SS][s C][s
U][s S][s T][s O][s M][SS][s S][s E][s T
]"                                          GM
•954 PRINTTAB(86)"1. [s R][s E][s G][s U]
[s L][s A][s R][SS][s C][s H][s A][s R][
s A][s C][s T][s E][s R][SS][s S][s E][s
 T]"                                        PN
•956 PRINTTAB(46)"2. [s U][s P][s P][s E]
[s R][SS][s A][s N][s D] LOWER [s C][s A
][s S][s E][SS][s S][s E][s T]"SPC(53)"3
. [s R][s E][s T][s U][s R][s N][SS][s T
][s O][SS][s M][s E][s N][s U]              ME
•960 PRINTTAB(87)"([s P][s R][s E][s S][s
 S][SS][s N][s U][s M][s B][s E][s R][SS
][s F][s O][s R][SS][s C][s H][s O][s I]
[s C][s E])":POKEBL,27                      CK
•962 WAITW,H:G=PEEK(631)-48:POKEW,.:IFG<H
ORG>3THEN962                                PC
•963 ONGGOTO965,966                         OA
•964 GOTO410                                CJ
•965 POKE685,208:SYS49213:CN=4:GOTO968      PI
•966 POKE685,216:SYS49213:CN=6              HO
•968 PRINT"[CLEAR]"TAB(242)". . . [s O][s
 K], SETS SWITCHED":FORTT=.TO1000:NEXT      EE
•970 O=.:GOSUB381:GOTO410                   AA
•999 REM ***MACHINE LANGUAGE***             OA
•1000 POKE53281,7:POKE53280,9:PRINT"[CLEA
R]";                                        HP
•1004 PRINT"[CLEAR]"SPC(162)"[c 2][RVSON]
[4" "]":PRINT"[c 2][RIGHT][RIGHT][RVSON]
 [RVSOFF]  [RVSON][sEP]  [c *]"             LP
•1006 PRINT"[RIGHT][RIGHT][RVSON][4" "][R
VSOFF]  [RVSON][4" "][c *]"                 PF
•1008 PRINT"[RIGHT][RIGHT][RVSON] [RVSOFF
]  [RVSON] [RVSOFF]  [RVSON]  [RVSOFF]
[RVSON] [c *][RVSOFF]  [RVSON][sEP]"        DN
•1010 PRINT"[RIGHT][RIGHT][RVSON] [RVSOFF
]  [RVSON] [RVSOFF]  [RVSON][4" "][RVSOF
F][sEP][RVSON] [c *][sEP] [sEP]  [c *]"     IN
```

```
•1012 PRINTTAB(5)"[c *][RVSON]  [RVSOFF][
sEP][RVSON] [RVSOFF]  [RVSON][c *] [RVSO
FF][c *][sEP][RVSON]  [RVSOFF]  [RVSON][
6" "]"                                     MJ
•1014 PRINTTAB(9)"[RVSON] [RVSOFF]  [RVSO
N]  [RVSOFF]  [RVSON][5" "][RVSOFF]  [RV
SON][6" "]"                                DG
•1016 PRINTTAB(13)"[RVSON] [RVSOFF]  [RVS
ON][SS] [RVSOFF]  [RVSON] [RVSOFF]  [RVS
ON] [RVSOFF]  [RVSON] [RVSOFF]  [RVSON][
4" "]"                                     MA
•1018 PRINTTAB(17)"[RVSON] [RVSOFF]  [RVS
ON] [RVSOFF]  [RVSON][SS][RVSOFF]  [RVSO
N] [RVSOFF]  [RVSON] [RVSOFF][3" "][RVSO
N][3" "][c *]"                             HK
•1020 PRINTTAB(23)"[RVSON][SS][RVSOFF]  [
RVSON] [RVSOFF]  [RVSON][3" "][RVSOFF] [
RVSON] [RVSOFF]  [RVSON] "                 OE
•1022 PRINTTAB(26)"[RVSON] [RVSOFF]  [RVS
ON] [RVSOFF][3" "][RVSON][3" "][RVSOFF][
sEP]"                                      BE
•1024 PRINTTAB(29)"[RVSON][5" "][RVSOFF]
 [RVSON][c *]":PRINTTAB(33)"[RVSON] [RVS
OFF]  [RVSON] "                            EJ
•1025 PRINTTAB(11)"BY MARIE COON &"SPC(25
)"THE AHOY! STAFF"                         NN
•1026 PRINT:PRINT:PRINTTAB(10)"[BLUE]ONE
MOMENT PLEASE[3"."]"                       HH
•1027 POKE52,124:POKE56,124:CLR:POKE53265
,27                                        HA
•1028 E$="BAD DATA":B=736:FORJ=.TO31:READ
A:POKEB+J,A:CK=CK+A:POKEB+J,A:NEXT         JH
•1030 IFCK<>5088THENPRINTE$:STOP:REM DATA
 1100                                      HI
•1035 B=49152:FORK=1TO4:CK=.               KJ
•1040 FORJ=.TO255:READA:POKEB+J,A:CK=CK+A
:NEXT                                      OB
•1045 READC:IFCK<>CTHENPRINTE$" IN BLOCK"
K"LINE"PEEK(64)*256+PEEK(63):STOP          EP
•1050 B=B+256:NEXT                         EP
•1055 CK=.:FORJ=.TO160:READA:POKEB+J,A:CK
=CK+A:NEXT                                 LD
•1060 IFCK<>20606THENPRINTE$" IN BLOCK 5"
:STOP                                      KK
•1065 POKE685,208:SYS49213:CLR:POKE686,1:
GOTO386                                    NN
•1085 DATA144,5,28,159,156,30,31,158,129,
149,150,151,152,153,154,155                KB
•1090 DATA208,133,156,223,220,158,159,222
,193,213,214,215,216,217,218,219           HJ
•1095 REM *** BLOCK 1 ***                  FD
•1100 DATA169,1,133,87,169,8,133,88,169,0
,133,57,133,58,160,3                       BB
•1102 DATA177,87,201,11,176,24,133,58,136
,177,87,133,57,160,0,177                   OM
•1104 DATA87,72,200,177,87,133,88,104,133
,87,160,3,208,226,165,87                   GL
•1106 DATA133,43,141,129,2,165,88,133,44,
141,130,2,96,173,14,220                    FK
•1108 DATA41,254,141,14,220,165,1,41,251,
133,1,169,0,133,87,133                     NF
•1110 DATA89,173,173,2,133,88,169,128,133
,90,162,8,160,0,177,87                     BO
•1112 DATA145,89,136,208,249,230,88,230,9
0,202,208,242,165,1,9,4                    IF
•1114 DATA133,1,173,14,220,9,1,141,14,220
,169,128,141,138,2,96                      OM
•1116 DATA173,14,220,41,254,141,14,220,16
5,1,41,251,133,1,169,0                     JK
•1118 DATA133,2,133,87,133,89,133,252,133
,254,173,173,2,133,88,169                  GL
•1120 DATA128,133,90,162,0,160,0,169,8,13
3,253,133,251,177,87,209                   LA
•1122 DATA89,240,2,230,252,202,240,32,200
,198,251,208,240,165,252,240               HD
•1124 DATA17,169,0,133,252,134,91,166,2,1
65,254,157,0,197,230,2                     ME
•1126 DATA166,91,230,254,169,8,208,211,23
0,88,230,90,198,253,208,216                JF
•1128 DATA165,1,9,4,133,1,173,14,220,9,1,
141,14,220,96,169                          LH
•1130 DATA0,133,87,133,89,169,216,133,88,
169,124,133,90,162,4,160,31556             HE
•1132 :                                    DI
•1134 REM *** BLOCK 2 ***                  EM
•1136 DATA0,177,87,145,89,136,208,249,230
,88,230,90,202,208,242,96                  AP
•1138 DATA169,0,133,87,133,89,169,216,133
,88,169,124,133,90,162,4                   DE
•1140 DATA160,0,177,89,145,87,136,208,249
,230,88,230,90,202,208,242                 JF
•1142 DATA96,169,0,133,251,133,253,141,17
0,2,141,171,2,169,25,133                   EI
•1144 DATA2,169,136,133,252,169,124,133,2
54,96,169,11,162,71,157,7                  GO
•1146 DATA216,202,208,250,169,32,162,71,1
57,7,4,202,208,250,169,8                   LH
•1148 DATA133,89,169,4,133,90,160,0,132,8
8,173,171,2,240,13,133                     DN
•1150 DATA87,169,40,56,237,171,2,133,91,7
6,69,194,132,87,140,168                    BF
•1152 DATA2,140,169,2,169,40,133,91,177,2
51,201,32,208,45,238,168                   MK
•1154 DATA2,200,198,91,208,242,169,145,16
4,88,145,89,198,2,208,6                    OA
•1156 DATA169,147,32,210,255,96,230,88,24
,165,251,105,40,144,4,230                  ED
•1158 DATA252,230,254,133,251,133,253,160
,0,240,193,201,34,208,3,238                LB
•1160 DATA168,2,173,168,2,240,17,132,87,1
69,157,164,88,145,89,230                   IC
•1162 DATA88,206,168,2,208,245,164,87,177
,253,41,15,205,167,2,240                   CN
•1164 DATA13,141,167,2,170,189,240,2,164,
88,145,89,230,88,173,170                   BK
•1166 DATA2,240,17,164,87,177,251,201,128
```

```
,176,32,206,170,2,164,88,34517                FC
•1168 :                                       DI
•1170 REM *** BLOCK 3 ***                     FJ
•1172 DATA169,210,208,15,164,87,177,251,2
01,128,144,15,238,170,2,164                   MD
•1174 DATA88,169,146,145,89,230,88,164,87
,177,251,41,127,201,34,208                    PO
•1176 DATA2,169,32,164,88,145,89,230,88,1
98,91,240,101,230,87,165                      AK
•1178 DATA88,201,60,144,16,200,169,34,145
,89,200,169,59,145,89,165                     HN
•1180 DATA87,141,171,2,96,164,87,177,251,
201,32,208,7,238,169,2                        IH
•1182 DATA208,215,208,154,173,169,2,240,3
1,240,147,173,170,2,240,11                    OH
•1184 DATA169,210,164,88,145,89,230,88,20
6,170,2,169,32,164,88,145                     AN
•1186 DATA89,230,88,206,169,2,208,245,164
,87,177,253,41,15,205,167                     FJ
•1188 DATA2,240,214,164,88,141,167,2,170,
189,240,2,145,89,230,88                       EI
•1190 DATA208,192,169,34,164,88,145,89,17
3,169,2,208,5,200,169,59                      EB
•1192 DATA145,89,198,2,208,6,169,59,200,1
45,89,96,24,165,251,105                       JB
•1194 DATA40,144,4,230,252,230,254,133,25
1,133,253,169,0,141,171,2                     OD
•1196 DATA141,170,2,96,169,1,141,25,208,1
62,130,160,12,173,18,208                      PE
•1198 DATA16,4,162,1,164,254,142,18,208,1
40,33,208,173,13,220,41                       IK
•1200 DATA1,240,3,76,49,234,76,188,254,16
9,0,133,89,165,209,133                        CF
•1202 DATA87,24,165,210,101,4,133,88,164,
211,177,209,72,170,177,87,33915               IE
•1204 :                                       DI
•1206 REM *** BLOCK 4 ***                     FC
•1208 DATA133,91,173,134,2,145,87,24,177,
209,105,128,145,209,230,89                    MK
•1210 DATA24,165,162,105,20,133,90,32,228
,255,208,21,165,90,197,162                    HL
•1212 DATA208,245,165,89,240,220,138,145,
209,165,91,145,87,198,89,240                  LP
•1214 DATA223,170,104,164,211,145,209,165
,91,145,87,96,32,233,194,138                  OD
•1216 DATA201,133,144,25,201,138,208,14,1
64,211,132,165,136,177,209,133                CN
•1218 DATA3,165,214,133,166,138,201,141,1
76,3,133,253,96,41,127,201                    EF
•1220 DATA33,144,5,173,134,2,145,87,173,1
41,2,201,4,240,6,138                          KO
•1222 DATA32,210,255,208,199,165,197,201,
20,208,244,173,134,2,133,87                   HE
•1224 DATA32,228,255,240,251,174,141,2,22
4,4,240,4,224,2,208,240                       GD
•1226 DATA162,15,221,224,2,240,8,202,208,
248,221,224,2,208,225,134                     KB
•1228 DATA88,169,0,133,89,169,216,133,90,
162,4,160,0,177,89,41                         LD
•1230 DATA15,197,87,208,4,165,88,145,89,1
36,208,241,230,90,202,208                     OL
•1232 DATA236,76,60,195,32,210,255,32,233
,194,138,201,133,144,5,201                    DK
•1234 DATA141,176,1,96,169,2,141,134,2,13
8,201,48,208,6,169,3                          BA
•1236 DATA141,134,2,138,41,127,201,13,208
,34,169,17,32,210,255,164                     JL
•1238 DATA211,177,209,201,32,208,4,169,14
5,208,201,169,157,32,210,255,35397            NO
•1240 :                                       DI
•1242 REM *** BLOCK 5 ***                     DP
•1244 DATA164,211,177,209,201,32,208,243,
169,29,208,184,201,17,240,12                  GE
•1246 DATA201,29,240,8,201,33,144,175,162
,29,176,1,138,32,210,255                      MI
•1248 DATA164,211,177,209,201,32,208,159,
24,138,105,128,208,150,169,0                  IL
•1250 DATA133,91,133,163,169,128,133,164,
160,3,165,3,24,10,38,91                       CI
•1252 DATA136,208,250,133,163,24,165,164,
101,91,133,164,24,160,30,162                  CM
•1254 DATA1,32,240,255,169,2,141,134,2,32
,196,195,169,69,133,87                        DK
•1256 DATA169,140,133,88,162,8,169,1,133,
91,160,8,169,0,133,92                         MJ
•1258 DATA177,87,201,48,240,6,165,92,5,91
,133,92,6,91,136,208                          BL
•1260 DATA239,165,92,145,163,230,163,208,
2,230,164,24,165,87,105,40                    IM
•1262 DATA133,87,144,2,230,88,202,208,205
,164,165,166,166,24,76,240,255                GE
•1595 REM ***DISPLAY***                       JB
•1600 POKE56576,5:GOSUB126:POKER+H,15:POK
ER,12:POKEA,39                                LE
•1602 POKEBL,PEEK(BL)AND239:PRINT"[CLEAR]
[BLACK]"TAB(131)"[s N][s E][s W][SS][s S
][s C][s R][s E][s E][s N][SS][s D][s I]
[s S][s P][s L][s A][s Y]"                    OH
•1603 PRINTTAB(84)"[s T]O DISPLAY SUCCESS
IVE SCREENS,                                  FI
•1604 PRINT"[4"[RIGHT]"]PRESS THE [s S][s
 P][s A][s C][s E] KEY.                       LJ
•1605 PRINTTAB(44)"[s P]RESS 'R' TO RETUR
N CONTROL TO                                  AH
•1606 PRINT"[4"[RIGHT]"]THE FORMAT SCREEN
 AND ACTIVATE                                 BI
•1607 PRINT"[4"[RIGHT]"]THE CURSOR.           DL
•1609 PRINTTAB(44)"[s A]NY OTHER KEY WILL
 RETURN TO THE                                AB
•1610 PRINT"[4"[RIGHT]"][s M]ENU, UNLESS
THE LAST SCREEN IS                            LG
•1611 PRINT"[4"[RIGHT]"]ON DISPLAY.  [s I
]N THIS CASE, ANY                             MN
•1612 PRINT"[4"[RIGHT]"]KEY BUT 'R' WILL
RETURN TO THE"SPC(10)"MENU.":POKEBL,27        EJ
•1614 WAITW,H:G=PEEK(631):POKEW,.:IFG<>32
```

IMPORTANT! Letters on white background are **Bug Repellent** line codes. **Do not enter them!** Pages 87 and 88 explain these codes and provide other essential information on entering ***Ahoy!*** programs. Refer to these pages **before** entering any programs!

```
THENPRINT"[CLEAR]":GOTO408                  OF
·3000 POKE56576,5:POKE648,136:POKE53272,3
7                                           DB
·3001 W=198:H=1:R=53280:A=53272:T=646       LP
·60000 PRINT"[CLEAR]":POKE646,X:SC=M:BC=Z
:GOTO408                                    FI
·60002 POKEA,37:PRINT"[CLEAR]"TAB(46)"NEW
 SCREEN STARTS AT"SPC(60)"PROGRAM LINE #
"                                           FI
·60004 PRINTTAB(46)"(HIT ANY KEY FOR DISP
LAY)":RETURN                                OP
·60006 FORJ=1TO 3                           HP
·60008 READB:CH=32768+8*B:PRINTCH:FORK=.T
O7:READA:POKECH+K,A:NEXT:NEXT               BJ
```

FIDGITS

FROM PAGE 39

```
·1 REM"[c A][30"[s *]"][c S]"               LE
·2 REM"[s B][11" "]FIDGITS[12" "][s B]"     HD
·3 REM"[s B]  BY WALTER E. MEYERS  1985[3
" "][s B]"                                  BN
·4 REM"[c Z][30"[s *]"][c X]"               JI
·5 POKE 56576,PEEK(56576) OR 3:POKE 53272
,21:POKE 648,4:PRINT "[CLEAR]":POKE 5328
1,15                                        KJ
·10 POKE 53280,15:DIM B$(26),NO(26),F$(4)
,M$(2):FOR I=1 TO 129:READ A$:NEXT          HJ
·15 S=54272:GOTO 1000                       IP
·20 IF N<8 THEN LN=0:COL=(N*4)+2:RETURN     BJ
·30 IF N<17 THEN NN=N-8:LN=8:COL=(NN*4)+2
:RETURN                                     FK
·40 NN=N-17:LN=16:COL=NN*4:RETURN           IN
·50 POKE 781,LN:POKE 782,COL:POKE 783,0:S
YS 65520:RETURN                             KJ
·60 FOR C=S TO S+24:POKEC,0:NEXT:RETURN     LO
·70 READ H,L,DR:POKE S+1,H:POKE S,L:POKES
+4,17                                       CJ
·80 FOR T=1 TO DR:POKE S+15,63:NEXT:POKE
S+4,20:RETURN                               NI
·90 N=INT(RND(0)*26)+1:RETURN               PM
·99 REM ---------------------- TITLE PAGE   NK
·100 PRINT"[CLEAR]":GOSUB 60:FOR I=1 TO 6
:PRINT:NEXT:FOR J=1 TO 3:FOR I=1 TO 4       PH
·110 PRINT F$(I) SPC(18) F$(I);:NEXT:NEXT
:LN=11:COL=18:GOSUB 50:PRINT B7$            DN
·120 M=1:LL=8:N=12:FOR I=1 TO 30:LN=6:COL
=18:GOSUB 50:PRINT M$(M)                    DB
·130 LN=LL:COL=12:GOSUB 50:PRINT WL$:COL=
23:GOSUB 50:PRINT WR$                       LC
·140 GOSUB 90:HI=INT(NO(N)/256):LO=NO(N)-
(HI*256):POKE S+5,0:POKE S+6,240            HJ
·150 POKE S+24,15:POKE S+1,HI:POKE S,LO:P
OKE S+4,17                                  AN
·160 FOR T=1 TO 50:NEXT:POKE S+4,16:LL=AB
S(17-LL):M=ABS(3-M):NEXT                    FO
·170 GOSUB 60:FOR T=1 TO 100:NEXT           PA
·299 REM ------------------- INSTRUCTIONS   HH
·300 PRINT "[CLEAR][DOWN][DOWN]":FOR I=1
TO 4:PRINT F$(I) SPC(18) F$(I);:NEXT:PRI
NT                                          IO
·310 PRINT"[c 4][5" "]FIDGITS ARE LIKE BA
BY BIRDS, ONLY[DOWN][4" "]SQUARER.  WHEN
 THEY GET";                                 PG
·320 PRINT" HUNGRY, THEY[DOWN][4" "]SING
FOR THEIR SUPPER.  TO QUIET THE[DOWN][4"
 "]SINGING";                                DA
·330 PRINT" FIDGIT, YOU FEED IT BY PRES-[
DOWN][4" "]SING THE KEY WITH THE SAME";     MN
·340 PRINT" LETTER AS[DOWN][4" "]THE FIDG
IT."                                        FI
·399 REM ---------------- LOAD CUST FONT    EJ
·400 FOR I=0 TO 32:READ A:POKE 49152+I,A:
NEXT                                        JE
·410 POKE 56334,PEEK(56334) AND 254:POKE
1,PEEK(1) AND 251:SYS 49152                 NE
·420 POKE 1,PEEK(1) OR 4:POKE 56334,PEEK(
56334) OR 1                                 IE
·430 READ A:IF A=999 THEN 450               JN
·440 FOR I=0 TO 7:READ B:POKE 14336+(A*8)
+I,B:NEXT:GOTO 430                          AH
·450 PRINT:PRINT "  [c 5][RVSON] HIT ANY
KEY WHEN READY TO CONTINUE."                EB
·460 GET A$:IF A$="" THEN 460               HJ
·470 PRINT CHR$(147):POKE 53272,(PEEK(532
72) AND 240) OR 14                          IL
·499 REM ------------------ SET UP SCREEN   CB
·500 GOSUB 60:POKES+5,9:POKE S+24,15:PRIN
T CHR$(147) SPC(6);                         CN
·510 RESTORE:FOR I=1 TO 7: PRINT HC$;B$(I
);"[RVSOFF][7"[UP]"]";:GOSUB 70:NEXT        CA
·520 PRINT:PRINT"[7"[DOWN]"]  ";            KM
·530 FOR I=8 TO 16:PRINT HC$;B$(I);"[RVSO
FF][7"[UP]"]";:GOSUB 70:NEXT:PRINT:PRINT
"[6"[DOWN]"]"                               KI
·540 FOR I=17 TO 22:PRINT HC$;B$(I);"[RVS
OFF][7"[UP]"]";:GOSUB 70:NEXT               AA
·550 PRINT HC$;B$(23);"[RVSOFF][7"[UP]"]"
;:FOR I=23 TO 25:GOSUB 70:NEXT              NK
·560 PRINT HC$;B$(24);"[RVSOFF][7"[UP]"]"
;:GOSUB 70:FOR J=25 TO 26                   OA
·570 PRINT HC$;B$(J);"[RVSOFF][7"[UP]"]";
:FOR I=1 TO 2:GOSUB 70:NEXT:POKE S+4, 20
:NEXT                                       KG
·580 FORI=1 TO 13:GOSUB 70:NEXT:GOSUB 60    CA
·598 POKE 808,234:REM
*NOTE: DISABLES STOP AND STOP/RESTORE       KN
·599 REM --------------------- MAIN LOOP    OP
·600 WL=99:GOSUB 90:RN=N:HI=INT(NO(N)/256
):LO=NO(N)-(HI*256)                         HK
·610 GOSUB 20:RL=LN:RC=COL:GOSUB 50:PRINT
 HO$:POKE S+5,0:POKE S+6,240                LH
```

```
•620 POKE S+24,15:POKE S+1,HI:POKE S,LO:P
OKE S+4,17:POKE 198,0                         IM
•630 GET N$:IF N$="" THEN 630                 LO
•640 N=ASC(N$):IF N=133 THEN 760              EP
•650 IF N<65 OR N>90 THEN 630                 EH
•660 N=N-64:IF WL<>99 THEN LN=WL:COL=WC:G
OSUB 50:PRINT HC$                             AE
•670 IF N=RN THEN LN=RL:COL=RC:GOTO 690       FI
•680 GOSUB 20:WL=LN:WC=COL:GOSUB 50:PRINT
 HS$:GOTO 630                                 NM
•690 GOSUB 50:PRINT HC$:SC=SC+1:POKE S+4,
16:IF SC<10 THEN 600                          CB
•700 GOSUB 60:GOSUB 900:PRINT:GOSUB 950:P
OKE 808,237:POKE 198,0                        BG
•710 GET A$:IF A$="" THEN 710                 JP
•720 IF A$<>"Q" THEN SC=0:GOTO 500            KC
•739 REM ------------------------ END GAME    CK
•740 GOSUB 900:PRINT:GOSUB 980:LN=3:FOR I
=1 TO 21:POKE 781,LN:POKE 782,13              LK
•750 POKE 783,0:SYS 65520:PRINT WL$:FOR T
=1 TO 100:NEXT:LN=ABS(5-LN):NEXT              HJ
•760 POKE 808,237:POKE 53272,21:POKE 648,
4:PRINT "[CLEAR][c 7]":POKE 53281,6:POKE
 53280,14                                     HB
•770 GOSUB 60:END                             LF
•899 REM --------------------- NEST SBRTN     KO
•900 PRINT"[CLEAR][DOWN][DOWN]":PRINTTAB(
18) B1$:PRINTTAB(18) B2$:PRINTTAB(16) LE
FT$(NE$,3);                                   IA
•910 PRINT B3$ LEFT$(NE$,3):PRINTTAB(13)
LEFT$(NE$,6) B4$ LEFT$(NE$,6)                 MK
•920 PRINTTAB(11) LEFT$(NE$,6)"[RVSOFF]"
B5$ LEFT$(NE$,6):PRINTTAB(10) LEFT$(NE$,
4);                                           JL
•930 PRINT "[RVSOFF]" B6$ LEFT$(NE$,4):PR
INTTAB(8) NE$:PRINTTAB(9) LEFT$(NE$,22)       NJ
•940 PRINTTAB(10) LEFT$(NE$,20):PRINTTAB(
11) LEFT$(NE$,18):RETURN                      GN
•949 REM ------------------ BANNER SBRTNS     EA
•950 PRINTTAB(9)"[RVSON][RED] [s G][s H][
c A][s S][c A][s I]  [c A][s I][s C][s D
][s U][s I][s A][s A][s G][s H] [s A] ":
PRINTTAB(9)"[RVSON] [c M][c H][c Q][s S]
 [s B][s B]  [c Q][s Z][c M][c H][s B][c
S][c Q][c W][c M][c H] [s B] "                FD
•960 PRINTTAB(9)"[RVSON] [s E][s F][c Z][
s S][s X][s X]  [s X][s X][s E][s F][s J
][s K][s X][s X][s E][s F] [s W] ":PRINT
TAB(6)"[c 4][DOWN]HIT '[BLUE]Q[c 4]' TO
QUIT";                                        DB
•970 PRINT" OR ANY OTHER":PRINTTAB(12)"KE
Y TO PLAY AGAIN.":POKE 198,0:RETURN           GE
•980 PRINTTAB(11)"[RVSON][RED] [c A][s I]
 [s A][s A][c A][s S]  [c A][s I][s A][s
A][c A][s S] [s A] ":    PRINTTAB(11) "
[RVSON] [c Q][s Z][s M][s N][c Q][s S]
 [c Q][s Z][s M][s N][c Q][s S] [s B] "       KA
•990 PRINTTAB(11) "[RVSON] [c Z][s K][s E
][s F][c Z][s S]  [c Z][s K][s E][s F][c
 Z][s S] [s W] ":RETURN                       DL
•999 REM ----------------------- STRINGS      HH
•1000 HC$="[4" "][DOWN][4"[LEFT]"][WHITE]
[s U][s C][s C][s I][DOWN][4"[LEFT]"][s
B][BLACK][RVSON]**[WHITE][RVSOFF][s B][D
OWN][4"[LEFT]"][s B][c 1][RVSON]$%[WHITE
][RVSOFF][s B][DOWN][4"[LEFT]"][s J][c R
][c R][s K][DOWN][4"[LEFT]"]"                 FN
•1010 HO$="[WHITE][s U][s C][s C][s I][DO
WN][4"[LEFT]"][s B][RED][RVSON][sEP][c *
][RVSOFF][WHITE][s B][DOWN][4"[LEFT]"][s
 B][RED][c *][sEP][WHITE][s B][DOWN][4"[
LEFT]"][s J][c R][c R][s K][DOWN][4"[LEF
T]"] [s -][s -] "                             NH
•1015 HS$="[4" "][DOWN][4"[LEFT]"][WHITE]
[RVSON]9[RVSOFF][c P][c P][RVSON]0[DOWN]
[4"[LEFT]"]1[BLACK]22[WHITE]3[DOWN][4"[L
EFT]"]4[c 1]$%[WHITE]5[DOWN][4"[LEFT]"][
RVSOFF][s J][c R][c R][s K][DOWN][4"[LEF
T]"]"                                         CP
•1020 B$(1)="[RVSON][BLACK]&[s U][s I]'[D
OWN][4"[LEFT]"] [c Q][c W] [DOWN][4"[LEF
T]"] [s X][s X] "                             BN
•1030 B$(2)="[RVSON][RED]&[c A][s I]'[DOW
N][4"[LEFT]"] [c Q][s Z] [DOWN][4"[LEFT]
"] [c Z][s K] "                               CL
•1040 B$(3)="[RVSON][PURPLE]&[s U][s I]'[
DOWN][4"[LEFT]"] [s B]  [DOWN][4"[LEFT]"
] [s J][s K] "                                MJ
•1050 B$(4)="[RVSON][BLUE]&[c A][s I]'[DO
WN][4"[LEFT]"] [s B][s B] [DOWN][4"[LEFT
]"] [c Z][s K] "                              GH
•1060 B$(5)="[RVSON][c 1]&[c A][s S]'[DOW
N][4"[LEFT]"] [c Q][s S] [DOWN][4"[LEFT]
"] [c Z][s S] "                               LD
•1070 B$(6)="[RVSON][c 2]&[c A][s S]'[DOW
N][4"[LEFT]"] [c Q][s S] [DOWN][4"[LEFT]
"] [s X]  "                                   EL
•1080 B$(7)="[RVSON][c 3]&[s U][s I]'[DOW
N][4"[LEFT]"] [s B][c S] [DOWN][4"[LEFT]
"] [s J][s K] "                               CM
•1090 B$(8)="[RVSON][c 5]&[s A][s A]'[DOW
N][4"[LEFT]"] [c Q][c W] [DOWN][4"[LEFT]
"] [s X][s X] "                               OO
•1100 B$(9)="[RVSON][GREEN]&[s C][s D]'[D
OWN][4"[LEFT]"] [c M][c H] [DOWN][4"[LEF
T]"] [s E][s F] "                             FL
•1110 B$(10)="[RVSON][RED]& [s A]'[DOWN][
4"[LEFT]"]  [s B] [DOWN][4"[LEFT]"] [s J
][s K] "                                      IH
•1120 B$(11)="[RVSON][BLACK]&[s A][s A]'[
DOWN][4"[LEFT]"] [c Q][s Z] [DOWN][4"[LE
FT]"] [s X][s X] "                            DM
•1130 B$(12)="[RVSON][PURPLE]&[s A] '[DOW
N][4"[LEFT]"] [s B]  [DOWN][4"[LEFT]"] [
c Z][s S] "                                   MD
•1140 B$(13)="[RVSON][BLUE]&[]'[DOWN][4"[
```

```
LEFT]"] 78 [DOWN][4"[LEFT]"] :; "          ON
•1150 B$(14)="[RVSON][GREEN]&[c A][s I]'[
DOWN][4"[LEFT]"] [s B][s B] [DOWN][4"[LE
FT]"] [s X][s X] "                          LE
•1160 B$(15)="[RVSON][c 1]&[s U][s I]'[DO
WN][4"[LEFT]"] [s B][s B] [DOWN][4"[LEFT
]"] [s J][s K] "                            PD
•1170 B$(16)="[RVSON][c 2]&[c A][s I]'[DO
WN][4"[LEFT]"] [c Q][s K] [DOWN][4"[LEFT
]"] [s X]  "                                EB
•1180 B$(17)="[RVSON][c 1]&[s U][s I]'[DO
WN][4"[LEFT]"] [s B][s B] [DOWN][4"[LEFT
]"] [s J][s Q] "                            NP
•1190 B$(18)="[RVSON][RED]&[c A][s I]'[DO
WN][4"[LEFT]"] [c Q][s Z] [DOWN][4"[LEFT
]"] [s X][s X] "                            MB
•1200 B$(19)="[RVSON][BLUE]&[s U][s I]'[D
OWN][4"[LEFT]"] [s J][s I] [DOWN][4"[LEF
T]"] [s J][s K] "                           NH
•1210 B$(20)="[RVSON][PURPLE]&[s G][s H]'
[DOWN][4"[LEFT]"] [c M][c H] [DOWN][4"[L
EFT]"] [s E][s F] "                         BE
•1220 B$(21)="[RVSON][c 5]&[s A][s A]'[DO
WN][4"[LEFT]"] [s B][s B] [DOWN][4"[LEFT
]"] [s J][s K] "                            DJ
•1230 B$(22)="[RVSON][GREEN]&[s A][s A]'[
DOWN][4"[LEFT]"] [s B][s B] [DOWN][4"[LE
FT]"] [c G][c N] "                          HL
•1240 B$(23)="[RVSON][RED]&[UPARROW][BACK
ARROW]'[DOWN][4"[LEFT]"] 78 [DOWN][4"[LE
FT]"] <> "                                  MB
•1250 B$(24)="[RVSON][BLACK]&[s A][s A]'[
DOWN][4"[LEFT]"] [s V][s Z] [DOWN][4"[LE
FT]"] [s X][s X] "                          NN
•1260 B$(25)="[RVSON][c 2]&[s A][s A]'[DO
WN][4"[LEFT]"] [s M][s N] [DOWN][4"[LEFT
]"] [s E][s F] "                            DF
•1270 B$(26)="[RVSON][BLUE]&[s R]6'[DOWN]
[4"[LEFT]"] [c Y][c U] [DOWN][4"[LEFT]"]
 [c O][s S] "                               EE
•1280 B1$="[WHITE][s U][s *][s *][s I]":B
2$="[WHITE][s B][BLACK][RVSON]**[WHITE][
RVSOFF][s B][c 2]":B3$="[RVSOFF][WHITE][
s B][c 1][RVSON]$%[WHITE][RVSOFF][s B][c
 2]":B4$="[RVSOFF][WHITE][s J][c R][c R]
[s K][c 2]"                                 DN
•1290 B5$="  [RVSON][RED]&  '[RVSOFF][c 2
]  ":B6$="[5" "][RVSON][RED][4" "][RVSOF
F][c 2][5" "]":NE$="[RVSON]())[3"("]))()
())()())()(())"                             LA
•1300 B7$="[GREEN][RVSON][sEP]  [c *][DOW
N][4"[LEFT]"][4" "][DOWN][4"[LEFT]"][4"
"]"                                         HJ
•1310 WL$="[WHITE][5" "][DOWN][5"[LEFT]"]
[c A][3"[s *]"][s I][DOWN][5"[LEFT]"][s
J][c R]  [s B][DOWN][4"[LEFT]"][s J][c R
] [c Z][DOWN][3"[LEFT]"][s J][s *][s *][
DOWN][3"[LEFT]"][3" "]"                     JC
```

```
•1320 WR$="[WHITE][5" "][DOWN][5"[LEFT]"]
[s U][3"[s *]"][c S][DOWN][5"[LEFT]"][s
B]  [c R][s K][DOWN][5"[LEFT]"][c X] [c
R][s K][DOWN][4"[LEFT]"][s *][s *][s K][
DOWN][3"[LEFT]"][3" "][c 4]"                KP
•1330 F$(1)="[RED][s U][s I][s S] [c S]
[s S][3" "]":F$(2)="[s +] [c S][s U][c W
][s U][c S][c S][s +][s U][s I]":F$(3)="
[c E] [c E][s J][c E][s J][c W][c E][s J
][s J][s I]"                                HA
•1340 F$(4)="[5" "][s J][s K]  [s J][s K]
":M$(2)="[c 1][RVSON][c *][RVSOFF]  [RVS
ON][sEP][DOWN][5"[LEFT]"][WHITE][RVSOFF]
[s U][RVSON][c 1] [c *][sEP] [RVSOFF][WH
ITE][s I]"                                  GM
•1350 M$(1)="[c 1] [RVSON][sEP][c *][RVSO
FF] [DOWN][5"[LEFT]"][WHITE][s U][RVSON]
[c 1][sEP]  [c *][WHITE][RVSOFF][s I][DO
WN][6"[LEFT]"][s B]  [BLACK][s Q][WHITE]
 [s B][DOWN][6"[LEFT]"][s B][4" "][s B][
DOWN][6"[LEFT]"][s J][s *][c S][c A][s *
][s K]"                                     IF
•1399 REM ------------- NOTE FREQUENCIES   BK
•1400 NO(1)=4291:NO(2)=4547:NO(3)=4818:NO
(4)=5103:NO(5)=5407:NO(6)=5728              DH
•1410 NO(7)=6070:NO(8)=6431:NO(9)=6813:NO
(10)=7218:NO(11)=7648:NO(12)=8102           AE
•1420 NO(13)=8584:NO(14)=9095:NO(15)=9634
:NO(16)=10208:NO(17)=10815:NO(18)=11458     PC
•1430 NO(19)=12139:NO(20)=12861:NO(21)=13
625:NO(22)=14436:NO(23)=15294               PN
•1440 NO(24)=16204:NO(25)=17167:NO(26)=18
188                                         LG
•1450 GOTO 100                              CF
•1999 REM -------------------- MUSIC DATA   GH
•2000 DATA 33,136,24,33,136,24,50,61,24     HB
•2010 DATA 50,61,24,56,100,24,56,100,24     DN
•2020 DATA 50,61,48,44,194,24,44,194,24     IH
•2030 DATA 42,63,24,42,63,24,37,162,10      MA
•2040 DATA 37,162,10,37,162,10,37,162,10    IP
•2050 DATA 33,136,48,50,61,24,50,61,24      NA
•2060 DATA 44,194,48,42,63,24,42,63,24      AA
•2070 DATA 37,162,48,50,61,10,50,61,10      KJ
•2080 DATA 50,61,24,44,194,48,42,63,24      IJ
•2090 DATA 42,63,24,37,162,48,33,136,24     FF
•2100 DATA 33,136,24,50,61,24,50,61,24      LK
•2110 DATA 56,100,24,56,100,24,50,61,48     EH
•2120 DATA 44,194,24,44,194,24,42,63,24     FA
•2130 DATA 42,63,24,37,162,24,37,162,24     OE
•2140 DATA 33,136,48                        FF
•2999 REM --------------------- COPY DATA   JE
•3000 DATA 169,0,133,251,133,253,169,208    HB
•3010 DATA 133,252,169,56,133,254,162,16    FI
•3020 DATA 160,0,177,251,145,253,136,208    GD
•3030 DATA 249,230,252,230,254,202,208      AD
•3040 DATA 240,96                           LN
•3099 REM -------------- CUSTOM FONT DATA   NH
•3155 DATA155,255,255,255,0,0,12,12,12      JL
```

```
•3156 DATA156,255,255,255,255,255,255,255
,255                                          IO
•3157 DATA157,255,255,255,7,3,33,48,48        MC
•3158 DATA158,255,255,255,12,12,12,12,12      AP
•3159 DATA159,255,255,255,48,48,48,48,48      DD
•3164 DATA164,63,113,57,31,15,7,3,1           NN
•3165 DATA165,252,142,156,248,240,224,192
,128                                          PH
•3166 DATA166,15,15,7,15,31,63,127,255        CO
•3167 DATA167,240,240,224,240,248,252,254
,255                                          EI
•3168 DATA168,49,155,206,236,63,115,206,7
2                                             FG
•3169 DATA169,140,217,115,55,252,206,115,
18                                            ME
•3170 DATA170,126,231,0,60,126,60,0,0         AE
•3175 DATA175,0,0,0,0,0,0,7,15                IL
•3176 DATA176,0,0,0,0,0,0,224,240             FC
•3177 DATA177,28,24,24,12,12,24,48,48         DN
•3178 DATA178,60,102,195,24,60,60,60,24       FB
•3179 DATA179,56,24,24,48,48,24,12,12         AO
•3180 DATA180,24,28,14,7,7,14,28,24           DG
•3181 DATA181,24,56,112,224,224,112,56,24     JE
•3182 DATA182,255,255,255,3,3,135,135,15      NN
•3183 DATA183,12,12,12,12,12,12,12,12         PI
•3184 DATA184,48,48,48,48,48,48,48,48         HD
•3185 DATA185,0,0,0,0,0,0,3,7                 HC
•3186 DATA186,12,12,12,12,12,255,255,255      BM
•3187 DATA187,48,48,48,48,48,255,255,255      BE
•3188 DATA188,12,12,132,192,224,255,255,2
55                                            PO
•3190 DATA190,48,48,33,3,7,255,255,255        FP
•3193 DATA193,255,255,255,195,195,195,195
,195                                          BI
•3194 DATA194,195,195,195,195,195,195,195
,195                                          CI
•3195 DATA195,255,255,255,252,252,252,252
,252                                          HK
•3196 DATA196,255,255,255,63,63,63,63,63      DL
•3197 DATA197,252,252,252,252,252,255,255
,255                                          NA
•3198 DATA198,63,63,63,63,63,255,255,255      CH
•3199 DATA199,255,255,255,128,128,252,252
,252                                          AO
•3200 DATA200,255,255,255,1,1,63,63,63        CM
•3201 DATA201,255,255,255,15,7,131,195,19
5                                             JI
•3202 DATA202,195,195,193,224,240,255,255
,255                                          OH
•3203 DATA203,195,195,131,7,15,255,255,25
5                                             AA
•3205 DATA205,195,195,225,240,248,252,252
,252                                          LD
•3206 DATA206,195,195,135,15,31,63,63,63      PC
•3209 DATA209,3,3,3,1,0,241,251,255           DJ
•3210 DATA210,255,255,255,192,192,255,255
,255                                          BN
•3211 DATA211,255,255,255,3,3,255,255,255     JM
•3213 DATA213,255,255,255,240,224,193,195
,195                                          ND
•3214 DATA214,195,195,225,240,240,225,195
,195                                          ID
•3215 DATA215,255,195,129,129,195,255,255
,255                                          PG
•3216 DATA216,195,195,195,195,195,255,255
,255                                          NC
•3218 DATA218,195,195,135,15,15,135,195,1
95                                            PO
•3229 DATA229,195,225,240,248,252,255,255
,255                                          IM
•3234 DATA234,195,135,15,31,63,255,255,25
5                                             OL
•3235 DATA235,195,195,195,192,192,195,195
,195                                          HE
•3236 DATA236,255,255,255,195,195,195,195
,195                                          JM
•3237 DATA237,195,195,195,192,192,255,255
,255                                          BB
•3238 DATA238,255,255,255,3,3,195,195,195     KI
•3240 DATA240,255,255,255,192,192,195,195
,195                                          IF
•3241 DATA241,195,195,195,0,0,255,255,255     GA
•3242 DATA242,255,255,255,0,0,195,195,195     ID
•3243 DATA243,195,195,195,3,3,195,195,195     ML
•3247 DATA247,255,254,254,252,252,248,248
,240                                          LB
•3248 DATA248,15,31,31,63,63,127,127,255      OE
•3249 DATA249,240,225,225,192,192,255,255
,255                                          IB
•3999 DATA 999                                CP
```

SELECTACHROME

FROM PAGE 17

```
•10 REM**INIT**                               FN
•12 DIMT(15),N$(15),S$(15),T$(15)             OE
•14 C(0)=1:C(1)=3:C(2)=0:C(3)=6               HF
•16 FORI=0TO15:READS$(I):NEXT                 GD
•18 FORI=0TO15:READT$(I):T(I)=VAL(T$(I)):
NEXT                                          JD
•20 FORI=0TO15:READN$(I):NEXT                 GO
•30 REM**DATA**                               FL
•32 DATA0,1,2,3,4,5,6,7,8,9,10,11,12,13,1
4,15                                          FL
•34 DATA 144,5,28,159,156,30,31,158,129,1
49,150,151,152,153,154,155                    KB
•36 DATA BLACK,WHITE,RED,CYAN,PURPLE,GREE
N,BLUE,YELLOW                                 EA
•38 DATA ORANGE,BROWN,LT.RED,GRAY1,GRAY2,
LT.GREEN,LT.BLUE,GRAY3                        NB
•50 REM**DISPLAY**                            FE
•52 POKE53280,C(0):POKE53281,C(1)             EJ
```

```
•54 PRINTCHR$(147)CHR$(T(C(2))):PRINT      IA
•56 PRINTTAB(9)CHR$(18)" SELECTACHROME ":
 PRINT:PRINT:PRINT                         JE
•58 PRINT"CONTROL"TAB(12)"COLOR"TAB(25)"C
 OMMAND":PRINT                             KM
•60 PRINT:PRINT"F1 BORDER =";              LN
•62 PRINTC(0)TAB(15)N$(C(0))TAB(25);       NC
•64 PRINT"POKE53280,"S$(C(0))              NC
•66 PRINT:PRINT"F3 SCREEN =";              LH
•68 PRINTC(1)TAB(15)N$(C(1))TAB(25);       IK
•70 PRINT"POKE53281,"S$(C(1))              OI
•72 PRINT:PRINT"F5 TEXT1  =";              JC
•74 PRINTC(2)TAB(15)N$(C(2))TAB(25);       KK
•76 PRINT"PRINTCHR$(";                     AI
•78 PRINTT$(C(2))")"                       KK
•80 PRINTCHR$(T(C(3)))                     BM
•82 PRINT"F7 TEXT2  =";                    KH
•84 PRINTC(3)TAB(15)N$(C(3))TAB(25);       HK
•86 PRINT"PRINTCHR$(";                     AI
•88 PRINTT$(C(3))")"                       KH
•90 PRINT:PRINT:PRINT" HIT FUNCTION KEYS
 TO CHANGE COLORS"                         EL
•100 REM**CHANGE COLORS**                  NP
•102 GETK$:IFK$=""THEN102                  GK
•104 K=ASC(K$)-133:IFK<0ORK>7THEN102       GM
•106 M=INT(K/4):N=K-M*4                    KN
•108 IFM=0THENC(N)=C(N)+1:IFC(N)>15THENC(
 N)=0                                      BM
•110 IFMTHENC(N)=C(N)-1:IFC(N)<0THENC(N)=
 15                                        EC
•112 GOTO50                                PD
```

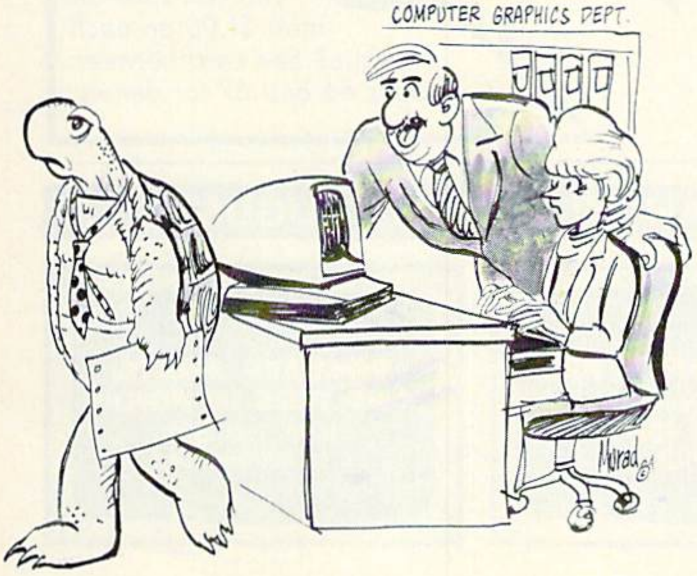
GATORS N SNAKES

FROM PAGE 41

```
•5 PX=53281                                HI
•10 L(0)=1350:L(1)=1510:L(2)=1670:L(3)=18
 30:COL=54272:LOC=L(0):K=0:LV=1:AG=0       GK
•20 SID=54272:SH=SID+1:SL=SH+3:W1=33:W2=3
 2:PRINT"[CLEAR][YELLOW][6"[DOWN]"]":POKE
 PX,11:POKEPX-1,0                          NN
•25 FORI=SIDTOSID+24:POKEI,0:NEXT:POKE542
 96,15:POKE54277,0:POKESH+5,240            MC
•30 POKE56,39:C=PEEK(53279):POKE50002,0    LK
•40 SN$="[6" "]G A T O R S  N  S N A K E
 S"                                        DJ
•45 FORI=1TO34:PRINTRIGHT$(SN$,I);"[UP]":
 FORT=1TO60:NEXT:POKESL,129:POKESH,7:POKE
 SID,255                                   PB
•47 NEXT:POKESL,128                        MJ
•50 PRINTTAB(12)"[3"[DOWN]"]BY JIM SANDER
 S"                                        MC
•60 PRINTTAB(14)"[DOWN][DOWN]ONE MOMENT[3
 "."][HOME]":GOSUB450:REM READ SPRITES/CH
 ARSET/ML                                  IO
•70 PRINT"[CLEAR]":K=0:SC=0:C=PEEK(53279)
 :POKE50002,0:POKE49203,11-LV:HT=0         CE
•80 POKE53265,PEEK(53265)AND239:POKEPX,11  GA
•90 PRINT"[WHITE][RVSON] SCORE ";TAB(37)"
 [c 2][RVSON]  [WHITE]"                    FM
•100 PRINT"[RVSON][7" "]";TAB(37)"[c 2]
 [WHITE]"                                  CL
•110 PRINT"[RVSON][3" "]HI  ";TAB(37)"[c
 2]  [WHITE]"                              LI
•120 PRINT"[RVSON] SCORE ";TAB(37)"[c 2]
  [WHITE]"                                 GA
•130 PRINT"[RVSON][7" "]";TAB(37)"[c 2]
 [WHITE]"                                  CL
•140 PRINT"[RVSON]G[6" "]";TAB(37)"[RVSOF
 F][YELLOW][s V][RVSON][YELLOW] [WHITE]"   JE
•150 PRINT"[RVSON]A[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 GN
•160 PRINT"[RVSON]T[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 LM
•170 PRINT"[RVSON]O[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 HI
•180 PRINT"[RVSON]R[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 LK
•190 PRINT"[RVSON]S[6" "]";TAB(37)"[RVSOF
 F][YELLOW][s V][RVSON][YELLOW] [WHITE]"   LA
•200 PRINT"[RVSON][7" "]";TAB(37)"[c 2]
 [WHITE]"                                  CL
•210 PRINT"[RVSON]N[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 LG
•220 PRINT"[RVSON][7" "]";TAB(37)"[c 2]
 [WHITE]"                                  CL
•230 PRINT"[RVSON]S[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 LP
•240 PRINT"[RVSON]N[6" "]";TAB(37)"[RVSOF
 F][YELLOW][s V][RVSON][YELLOW] [WHITE]"   HN
•250 PRINT"[RVSON]A[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 GN
•260 PRINT"[RVSON]K[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 HH
•270 PRINT"[RVSON]E[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 JB
•280 PRINT"[RVSON]S[6" "]";TAB(37)"[c 2]
  [WHITE]"                                 LP
•290 PRINT"[RVSON][7" "]";TAB(37)"[RVSOFF
 ][YELLOW][s V][RVSON][YELLOW] [WHITE]"    HM
•300 PRINT"[RVSON]LEV."LV;TAB(37)"[c 2]
 [WHITE]"                                  JB
•310 PRINT"[RVSON][7" "]";TAB(37)"[c 2]
 [WHITE][HOME]"                            CA
•320 FORI=0TO4                             JC
•330 PRINTTAB(18)"[RVSON][GREEN][3"[s X]"
 ][RVSOFF][WHITE]"                         FM
•340 NEXT:PRINT                            FO
•350 FORI=0TO8                             JG
•360 PRINTTAB(18)"[RVSON][GREEN][3"[s X]"
 ][RVSOFF][WHITE]"                         FM
•370 NEXT:PRINT                            FO
•380 FORI=0TO6                             JE
•390 PRINTTAB(18)"[RVSON][GREEN][3"[s X]"
 ][RVSOFF][WHITE]"                         FM
•400 NEXT:POKE1910,160:POKECOL+1910,1:POK
```

```
ESP+21,255                                        GH
·410 IFAG=1ANDUD=1THENPRINT"[HOME]";TAB(1
)"[5"[DOWN]"][RVSON]";HS                          EG
·420 POKEL(0),155:POKEL(1),155:POKEL(2),1
55:POKEL(3),155                                   ND
·430 POKEL(0)+COL,7:POKEL(1)+COL,7:POKEL(
2)+COL,7:POKEL(3)+COL,7                           CH
·440 POKE53265,PEEK(53265)OR16:GOTO1150           IC
·450 SP=53248:POKE53281,11:POKE53280,0            EN
·460 POKESP+21,0:POKESP+23,0:POKESP+16,19
2                                                 PM
·470 C1=SP+39:POKEC1,1:POKEC1+1,1:POKEC1+
2,7:POKEC1+3,7:POKEC1+4,1:POKEC1+5,1              MO
·480 POKEC1+6,5:POKEC1+7,15                       CN
·490 FORI=1TO15STEP2:POKESP+I,100:NEXT            KG
·500 POKESP+0,75:POKESP+2,140:POKESP+4,10
5:POKESP+6,195:POKESP+8,225:POKESP+10,25
5                                                 ND
·510 POKESP+12,15:POKESP+14,45                    CA
·520 M1=12288:M2=12352:M3=12416:M4=12480:
M5=12544:M6=12608:M7=12672:M8=12736               MA
·530 FORI=0TO63:READA:POKEM1+I,A:NEXT             FJ
·540 FORI=0TO63:READA:POKEM2+I,A:NEXT             DC
·550 FORI=0TO63:READA:POKEM3+I,A:NEXT             DL
·560 FORI=0TO63:READA:POKEM4+I,A:NEXT             DM
·570 RESTORE                                      IO
·580 FORI=0TO63:READA:POKEM5+I,A:NEXT             DN
·590 FORI=0TO63:READA:POKEM6+I,A:NEXT             DG
·600 FORI=0TO63:READA:POKEM7+I,A:NEXT             FP
·610 FORI=0TO63:READA:POKEM8+I,A:NEXT             GA
·620 FORI=0TO7:POKE2040+I,192+I:NEXT              NO
·630 FORI=3TO15STEP2:POKE50000+I,5:NEXT:P
OKE50001,4                                        LK
·640 DATA0,1,0,0,2,0,0,4                          ED
·650 DATA0,0,12,0,0,28,0,0                        OG
·660 DATA28,0,0,14,0,0,7,0                        MJ
·670 DATA0,3,128,0,1,192,0,1                      CA
·680 DATA192,0,3,128,0,7,0,0                      PH
·690 DATA14,0,0,28,0,0,30,0                       JC
·700 DATA0,15,0,0,13,128,0,15                     IM
·710 DATA128,0,3,128,0,0,128,0                    NB
·720 DATA0,64,0,0,224,0,1,240                     IO
·730 DATA0,1,152,0,0,248,0,0                      CL
·740 DATA120,0,0,48,0,0,112,0                     BI
·750 DATA0,224,0,1,192,0,3,128                    NG
·760 DATA0,3,128,0,1,192,0,0                      CB
·770 DATA224,0,0,112,0,0,56,0                     HC
·780 DATA0,56,0,0,48,0,0,32                       CI
·790 DATA0,0,64,0,0,128,0,0                       ND
·800 DATA0,1,0,0,6,0,0,12                         FF
·810 DATA0,0,24,0,0,24,0,0                        MP
·820 DATA12,0,0,14,0,0,14,24                      LN
·830 DATA1,143,32,0,95,160,0,95                   PB
·840 DATA224,0,127,128,0,31,128,0                 HF
·850 DATA31,128,0,127,128,0,95,224                FH
·860 DATA0,223,160,0,223,176,0,22                 BG
·870 DATA176,0,15,0,0,6,0,0                       NE
·880 DATA0,96,0,0,240,0,13,104                    ND
·890 DATA0,13,251,0,5,251,0,7                     KL
·900 DATA250,0,1,254,0,1,248,0                    KF
·910 DATA1,248,0,1,254,0,7,250                    HO
·920 DATA0,5,154,128,20,241,128,24                EL
·930 DATA112,0,0,112,0,0,48,0                     DH
·940 DATA0,24,0,0,24,0,0,48                       BC
·950 DATA0,0,96,0,0,128,0,0                       CO
·960 DATA120,165,1,41,251,133,1,169,0,133
,251,133,253,169,40,133,252,169,208,133           LG
·970 DATA254,162,8,160,0,177,253,145,251,
136,208,249,230,252,230,254,202,208,240           PJ
·980 DATA165,1,9,4,133,1,88,96                    KC
·990 DATA2,4,200,59,59,200,4,2,64,32,215,
216,216,23,32,64                                  IC
·1000 DATA153,90,36,24,36,102,102,102,102
,102,102,36,24,36,90,153                          AE
·1010 DATA129,60,66,90,90,66,60,129               KP
·1020 GOTO1030                                    FH
·1030 FORI=0TO46:READA:POKE828+I,A:CK=CK+
A:NEXT                                            MO
·1040 IFCK<>6871THENPRINT"ERROR IN LINES
960-980":END                                      HP
·1050 SYS828                                      EF
·1060 FORI=11480TO11519:READA:POKEI,A:NEX
T                                                 MF
·1070 POKE53272,(PEEK(53272)AND240)OR10           MM
·1080 FORL=SIDTOSID+24:POKEL,0:NEXT               EP
·1090 POKESID+5,88:POKESID+6,195:POKESID+
24,15                                             AI
·1100 FORI=49152TO49152+94:READML:POKEI,M
L:CK=CK+ML:NEXT                                   KL
·1110 IFCK<>20502THENPRINT"ERROR IN LINES
 1710-1780":END                                   EO
·1120 RETURN                                      IM
·1130 SYS49152:POKESH,25:POKESID,77               PB
·1140 POKESID+4,W1:FORT=1TO10:NEXT:POKESI
D+4,W2:SYS49152:W1=33:W2=32:RETURN                JJ
·1150 V=PEEK(56321)AND15                          EH
·1160 SYS49152                                    KF
·1170 IFPEEK(50002)<>0THENGOSUB1440               EK
·1180 IFV=14THEN1270                              IN
·1190 SYS49152                                    KF
·1200 IFV=13THEN1310                              HH
·1210 SYS49152                                    KF
·1220 IFV=7THEN1350                               DE
·1230 SYS49152                                    KF
·1240 IFV=11THEN1400                              GN
·1250 SYS49152                                    KF
·1260 GOTO1150                                    FO
·1270 UP=PEEK(LOC-40)                             FM
·1280 IFUP<>32THEN1150                            HO
·1290 POKELOC-40,157:POKECOL+LOC-40,7:POK
ELOC,32:GOSUB1130:LOC=LOC-40:SYS49152             IM
·1300 SYS49152:GOTO1150                           CC
·1310 DWN=PEEK(LOC+40)                            JO
·1320 IFDWN<>32THEN1150                           NB
·1330 POKELOC+40,158:POKECOL+LOC+40,7:POK
ELOC,32:GOSUB1130:LOC=LOC+40:SYS49152             PA
```

```
•1340 SYS49152:GOTO1150                    CC
•1350 RT=PEEK(LOC+1)                       DF
•1360 IFRT=86THENHT=HT+1:H=1:GOTO1380      JI
•1370 IFRT<>32THEN1150                     HL
•1380 POKELOC+1,155:POKECOL+LOC+1,7:POKEL
OC,32:GOSUB1130:LOC=LOC+1:SYS49152         MG
•1390 SYS49152:SC=SC+LV:GOSUB1480:GOTO115
0                                          EC
•1400 LT=PEEK(LOC-1)                       CI
•1410 IFLT<>32THEN1150                     HF
•1420 POKELOC-1,156:POKECOL+LOC-1,7:POKEL
OC,32:GOSUB1130:LOC=LOC-1:SYS49152         DG
•1430 SYS49152:SYS49152:GOTO1150           DD
•1440 IFK=3THENPOKELOC,159:W1=129:W2=128:
GOSUB1130:POKELOC,32:K=0:LOC=L(K):GOTO16
20                                         GO
•1450 K=K+1:X=HT+K:POKELOC,159:W1=129:W2=
128:GOSUB1130:POKELOC,32:LOC=L(K):C=PEEK
(53279)                                    IK
•1460 IFK=>4THENK=0:LOC=L(K):GOTO1620      FO
•1470 RETURN                               IM
•1480 SYS49152:PRINT"[HOME]";TAB(1)"[DOWN
][DOWN][RVSON]";SC:SYS49152                DN
•1490 IFHT=4THENPOKE1301,86:POKE1501,86:P
OKE1701,86:POKE1901,86:GOTO1510            MB
•1500 GOTO1580                             FP
•1510 POKE1350,155:POKE1510,155:POKE1670,
155:POKE1830,155:LOC=1350:K=0:HT=0:LV=LV
+1                                         IJ
•1520 IFLV>10THEN1790                      MK
•1530 PRINTTAB(4)"[RVSON][19"[DOWN]"]";LV
:IFLV=2THENPOKESP+23,1                     LG
•1540 IFLV=4THENPOKESP+23,5                GO
•1550 IFLV=6THENPOKESP+23,19               LK
•1560 IFLV=8THENPOKESP+23,51               JI
•1570 X=0:H=0:GOSUB1700:GOTO1610           ID
•1580 IFH=0THEN1610                        BA
•1590 X=HT+K:LOC=L(K):H=0:POKEL(K),155     PH
•1600 IFK=>4THENX=0:LOC=L(K):GOTO1620      IP
•1610 RETURN                               IM
•1620 IFSC>HSTHENHS=SC:UD=1                MN
•1630 PRINT"[HOME]";TAB(1)"[5"[DOWN]"][RV
SON]";HS                                   KB
•1640 POKESP+21,0:PRINTTAB(12)"GAME"TAB(2
3)"OVER"                                   NK
•1650 PRINT:PRINTTAB(12)"PLAY"TAB(22)"TO
AGAIN?"                                    NJ
•1655 PRINTTAB(12)"[RVSON][DOWN]PRESS"TAB
(21)"[RVSON]FIRE BUTTON"                   CM
•1660 JV=PEEK(56321):FR=JVAND16:IFFR=16TH
EN1660                                     JB
•1670 AG=1:LV=1:POKESP+23,0:GOTO70         HD
•1700 FORI=15TO0STEP-1:POKE53280,I:NEXT:R
ETURN                                      KO
•1710 DATA 24,160,0,185,1,208,121,81,195,
153,1,208                                  EA
•1720 DATA 200,200,200,200,192,20,208,239
,173,1,208,201                             IM
•1730 DATA 200,208,3,32,67,192,56,160,16,
185,255,207                                GC
•1740 DATA 249,79,195,153,255,207,136,136
,136,136,192,0                             OP
•1750 DATA 208,239,162,10,160,0,200,208,2
53,202,208,250                             EB
•1760 DATA 173,31,208,141,82,195,96,160,0
,140,80,195                                AF
•1770 DATA 32,151,224,165,141,41,5,172,80
,195,153,81                                BI
•1780 DATA 195,200,200,140,80,195,192,16,
208,234,96                                 JP
•1790 POKESP+21,0:POKE53281,11:PRINT"[CLE
AR]"                                       KE
•1800 PRINT"[YELLOW][6"[DOWN]"][5"[RIGHT]
"]YOU ARE A REMARKABLE PLAYER[3"."]"       GG
•1810 PRINTTAB(11)"[DOWN][DOWN]CONGRATULA
TIONS!"                                    LG
•1820 END                                  IC
```



"He's our expert on turtle graphics."

PROGRAMMERS!

Ahoy! wishes to continue publishing the best Commodore-compatible programs available in magazines today. We're looking for games, utilities, educational, music, and graphics programs, and programs that completely defy description. If you have an original C-64, C-128, or VIC 20 program, in BASIC or machine language, we'd like to see it. Payment will be made upon acceptance, at competitive industry rates. You will also receive royalties based on the sale of our program disks.

Send a copy of your program on disk or tape, accompanied by a printout and documentation, to *Ahoy!*, 45 West 34th Street— Suite 407, New York, NY 10001. Include a self-addressed mailer with sufficient return postage affixed. (Program printouts not accompanied by a disk or tape will be returned unread; programs not accompanied by sufficient return postage will not be returned.)

Continued from page 5

ing educational programs for school systems. A representative of Trillium Publishing revealed that programming for the C-64 and VIC just wasn't profitable. If it took under a week to rewrite an Apple program to run on the Commodore, they'd do it. But develop a package specifically for the 64? Too expensive.

Another discussion uncovered why this is true. Though more Commodore computers have been sold than any other brand, most Commodore users don't buy software: they steal it. Even protected programs, like *The Print Shop* and *Ghostbusters*, are quickly broken into and illegally distributed. One speaker said these two are currently the most popular "pirate" programs. Although the discussion centered on piracy within corporations, the point was well made that the majority of programs are stolen by "home" computerists.

ADAPSO, the Association of Data Processing Service Organizations, is helping to standardize a hardware "lock and key" system that when used with the more respected computer systems (like IBM and compatibles, Apple, and CP/M computers) will prevent software theft. Although backup disks can be made easily, the software won't run without the hardware key.

Within a year, such systems will make it very attractive to write software for these computers. Unfortunately, nothing similar is planned for "home" computer systems. Instead, programmers will just move into the more lucrative "business" field.

The pirates have done such an effective job of ripping off programs that companies are being forced to abandon us or risk bankruptcy. Even a company like Broderbund, creators of *The Print Shop*, may be forced to leave the "home" market behind. According to a *Future Computing* survey of 40,000 computer users, software's price has no relationship to the incidence of piracy. It had been theorized that as prices came down piracy would stop. This is not borne out by the numbers. Price seems to have no effect on the amount of stealing.

What does seem to matter is the amount of documentation and vendor support needed to use the program. The more difficult a package is to use, the less likely it is to be pirated. A menu-driven program that needs little or no documentation is much more likely to be stolen, i.e. programs like *HomeWord* and *The Print Shop*. (Unfortunately, the very factors that make a package a best seller also make it a "best stealer.")

I am not going to make impassioned pleas to the pirates to stop their activities. I now realize that the situation is hopeless. Instead, I'd like to give all the legitimate software customers a bit of praise and advice.

First, thank you for being the honest people that you are. At least one person in the world respects you. Second, if there is a piece of software you've had your eye on, buy it. Quick! Also, send back the registration card and give some helpful comments if you can. It can't hurt to let the companies that stick with us know that they are appreciated.

I'd also like to conduct a survey to find out how home users view their computers. According to the gurus of SoftCon, you don't exist. But, if you'd like to prove the home market isn't dead, drop me a line in care of *Ahoy!* If you want to be included in an article I'm writing on the subject, please send your name, address, type of computer and how you use it to:

Cheryl Peterson
c/o *Ahoy!*—Home Computer Use Survey

If you don't mind being contacted for more information, include your telephone number. ☐

...COMING IN THE SEPTEMBER *AHOY!* (ON SALE AUGUST 6)...

TUMBLING DICE AND DATA STRUCTURES
TAKING CHANCES WITH ADVANCED PROGRAMMING METHODS

TABLE TOP TYPER
A VERSATILE WORD PROCESSOR FOR THE COMMODORE 64

ROCKETS, BOATS, AND PIGS IN POKES
USING SPRITES IN C-64 ACTION GAMES

DON'T MISS THE BOAT!



SUBSCRIBE TO AHOY!

☐ Send me twelve issues of AHOY! for $19.95 ($26.95 Canada & elsewhere). I understand that I will be billed later.

☐ Send me twenty-four issues of AHOY! for $37.95 ($49.95 Canada & elsewhere). I understand that I will be billed later.

NAME ______________________

ADDRESS ______________________

CITY ______________________

STATE __________ ZIP __________

READER SERVICE CARD

To request additional information on any product advertised or otherwise pictured in this issue of *Ahoy!* that is accompanied by a reader service number, circle the corresponding number below and mail this card. We will promptly forward your request to the designated companies.

101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130
131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190
191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 210
211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240
241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270
271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300

PLEASE ANSWER THE QUESTIONS BELOW.

Sex and age
A ☐ male B ☐ female age __________

Educational level completed
C ☐ elementary
D ☐ high school
E ☐ junior college
F ☐ college graduate
G ☐ masters degree
H ☐ phd

From which of the following sources did you obtain your copy?
K ☐ newsstand N ☐ subscription (mail)
L ☐ from a friend or family member
M ☐ other ______________________

If not currently a subscriber, do you plan to become one?
I ☐ yes J ☐ no

Name ______________________

Street ______________________

City __________ State __________ Zip __________

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 65 MT. MORRIS, ILL.

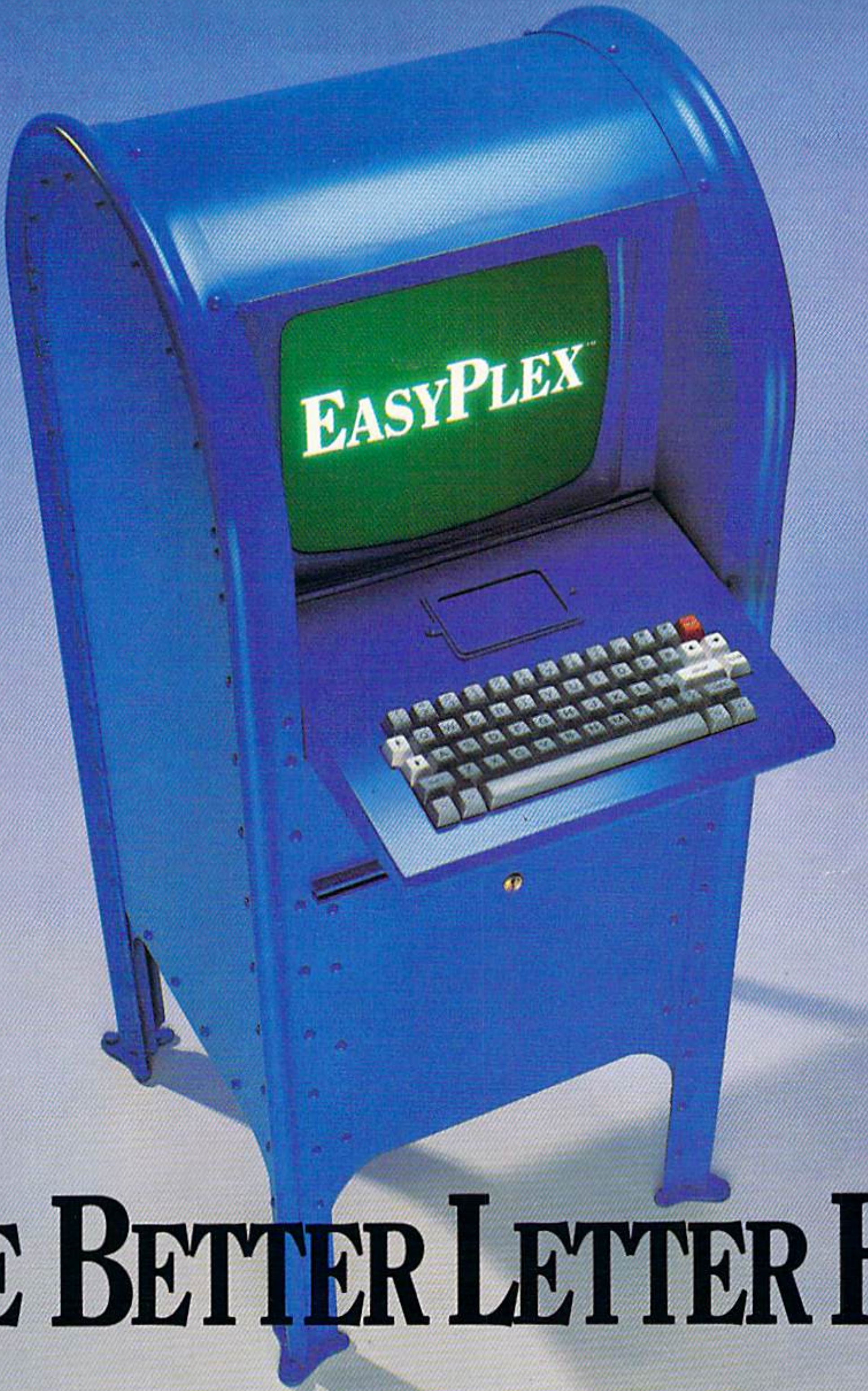
POSTAGE WILL BE PAID BY ADDRESSEE

Ahoy!

P.O. BOX 341
MT. MORRIS, ILL. 61054

PLACE
STAMP
HERE

AHOY! MAGAZINE
P.O. Box 2501
Clinton, Iowa 52735



THE BETTER LETTER BOX

Introducing EasyPlex.™ The new, easy-to-use electronic mail system from CompuServe.

Finally! Electronic Mail that's so easy to use you can start composing and sending messages the first time you get online.

Designed for various experience levels, EasyPlex has a menu mode with simple, easy-to-follow directions for beginners, and it lets experienced users save time by working in the prompt or command modes. With EasyPlex, you can compose, edit, send, file, and take advantage of sophisticated options previously available only with more expensive services.

CompuServe's EasyPlex lets friends and relatives, associations and club members communicate any time of the day or night. And small business owners, real estate professionals, insurance agents, lawyers, writers, etc. can communicate quickly and simply—either interoffice or interstate. "It's Easy." "Just Plex it!"

Best of all, EasyPlex is available to all CompuServe subscribers. And, along with EasyPlex, you get hundreds of valuable and entertaining computing options. Plus the assurance of belonging to the largest, fastest growing computer information service in the world and the premier supplier of business information to FORTUNE 500 companies.

Start communicating! To buy a CompuServe Subscription Kit, see your nearest computer dealer. To receive our informative brochure or to order direct, call or write:

CompuServe®

Information Services, P.O. Box 20212
5000 Arlington Centre Blvd., Columbus, Ohio 43220

800-848-8199

In Ohio, call 614-457-0802

An H&R Block Company

Reader Service No. 230



EXPAND IT
1
2
3
NAVARONE
NOW FOR YOUR COMMODORE 64
EXPAND YOUR COMMODORE 64™ WITH NAVARONE'S NEW CARTRIDGE EXPANDER
Plug in up to three cartridges at a time. Select the program you want. Even reset the computer independently of the power switch.
● Reduces cartridge swapping
● Built in reset button
● Saves wear & tear
NAVARONE
NAVARONE INDUSTRIES, Inc.
510 Lawrence Expressway, #800
Sunnyvale, CA 94086
(408) 985-2932
Reader Service No. 231
*Commodore 64 is a trademark of Commodore Electronics, LTD. respectively. This product is manufactured by Navarone Industries for use on the Commodore 64 computer and is not made, sponsored, authorized or approved by Commodore Electronics, LTD.